report no. 80HV007
july 1980

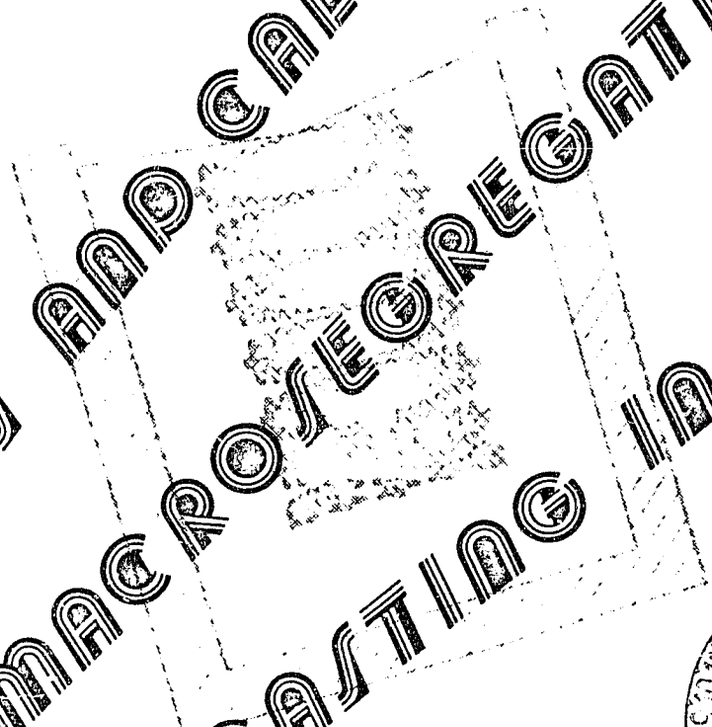
prepared for MSFC
under contract
no. NAS8-33573

N80-30480

(NASA-CR-161541) ANALYSIS AND CALCULATION OF MACROSEGREGATION IN A CASTING INGOT. MPS SOLIDIFICATION MODEL. VOLUME 2: SOFTWARE DOCUMENTATION Final Report (General Electric Co.) 84 p HC A05/MF A01 CSCL 11F G3/26 Unclas 28484

ANALYSIS AND CALCULATION OF MACROSEGREGATION IN A CASTING INGOT

volume II - software documentation



prepared by
the General Electric Co
huntsville operations
of the space division
huntsville alabama

GENERAL ELECTRIC

REPORT NO: 80HV007
DATE: JULY 1980

MPS SOLIDIFICATION MODEL

VOLUME II: SOFTWARE DOCUMENTATION

PREPARED FOR MSFC

UNDER CONTRACT NO. NAS8-33573

PREPARED BY: Anna L. Maples
Anna L. Maples
Program Manager

APPROVED BY:
Lowell A. Kelly, Manager
Huntsville Operations

TABLE OF CONTENTS

SECTION 1
INTRODUCTION

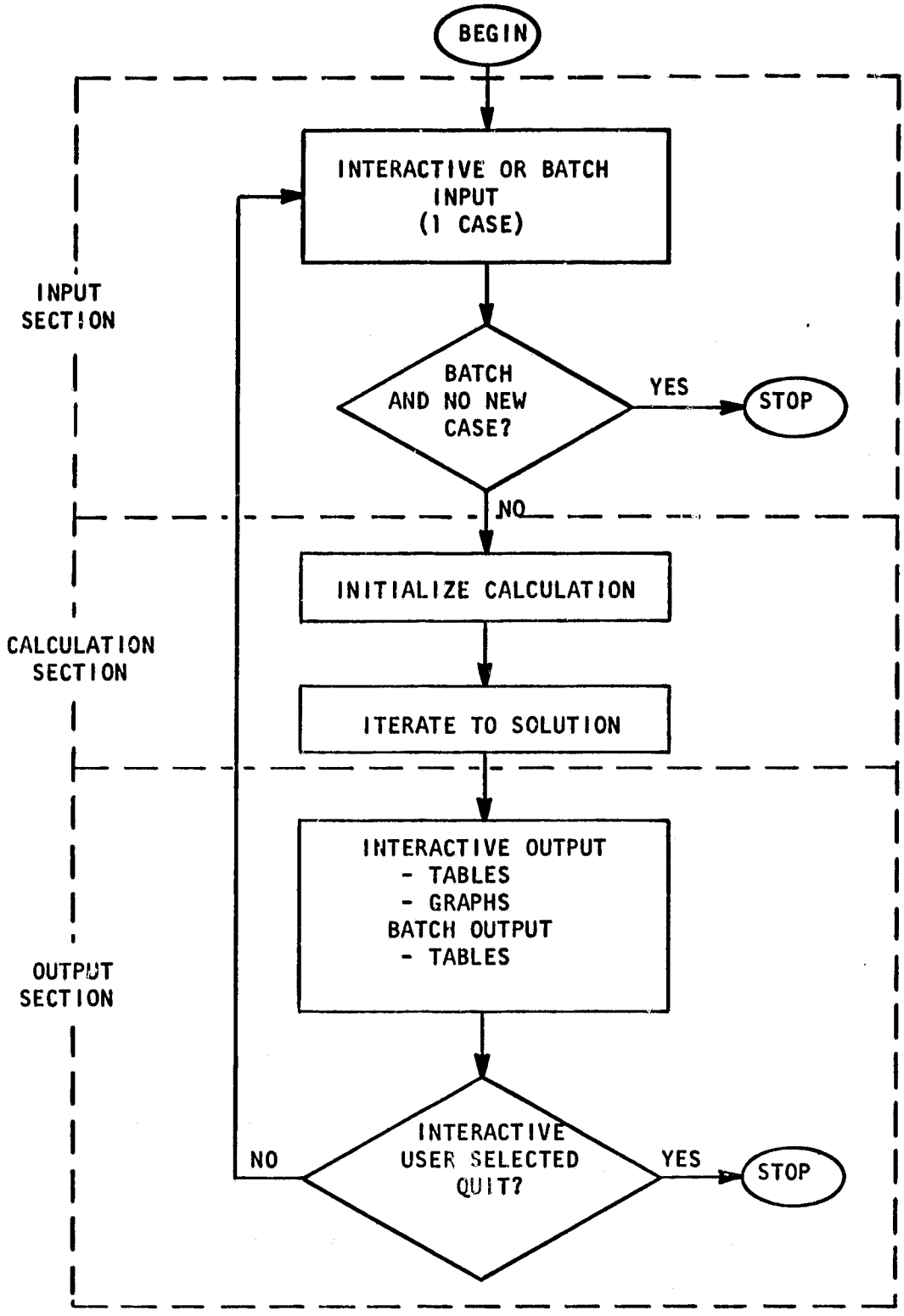
Volume II documents the software developed for the solidification model. This volume provides a link between the calculations described in Volume I and the FORTRAN code, primarily in the form of global flow diagrams and data structures. Considerable effort was expended in the design phase of the task to provide a code that is well structured, easily readable, and essentially self-documenting. Hence the minutia of the code need not be repeated here: indeed, such redundancy is best omitted because it can only provide an opportunity for discrepancy. A complete listing of the solidification code is in Appendix A.

It is assumed that the reader of this volume is familiar with the calculation described by Volume I, with the program operating characteristics as described by Volume III, and with the FORTRAN language.

SECTION 2

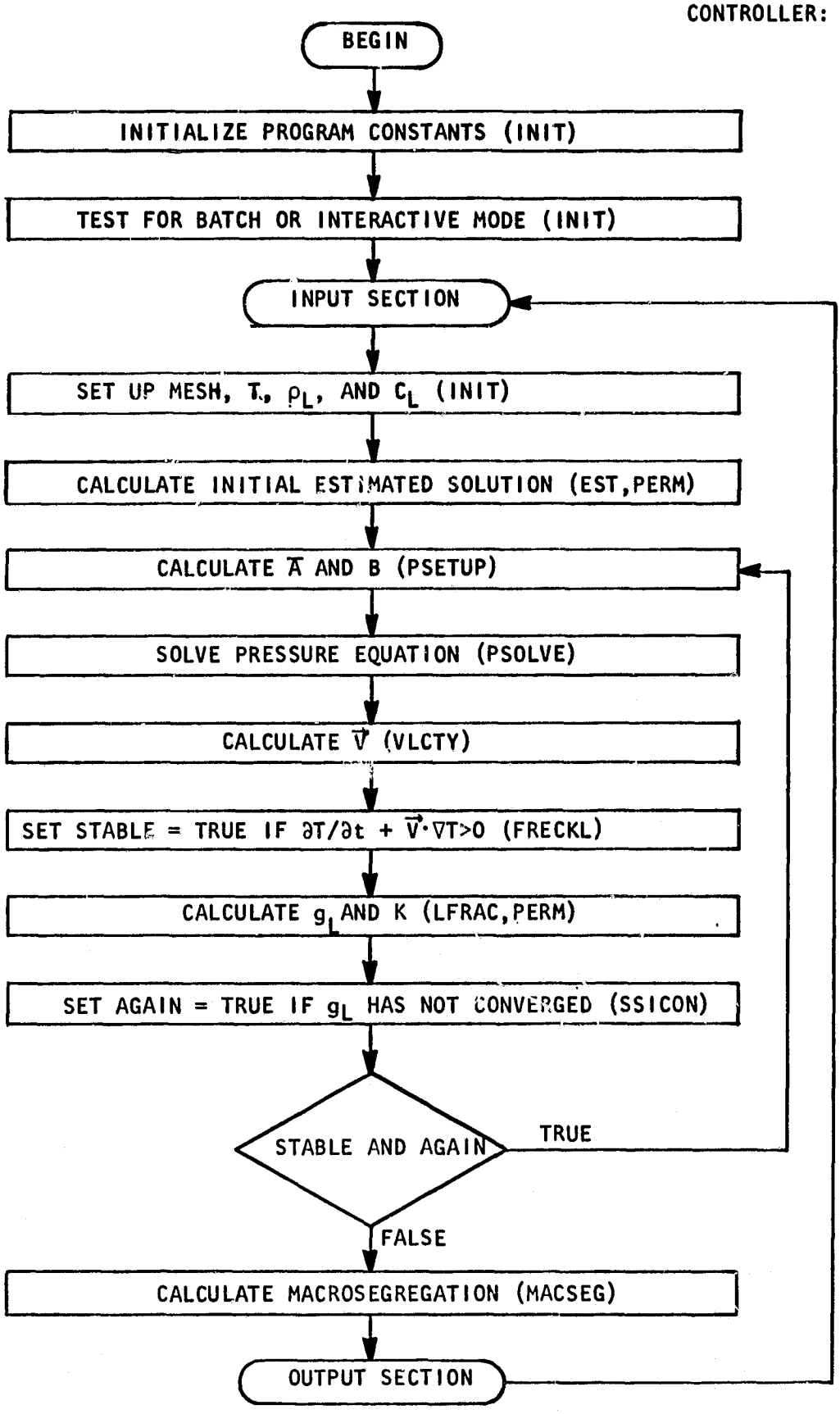
FLOW DIAGRAMS

2.1 GLOBAL FLOW DIAGRAM



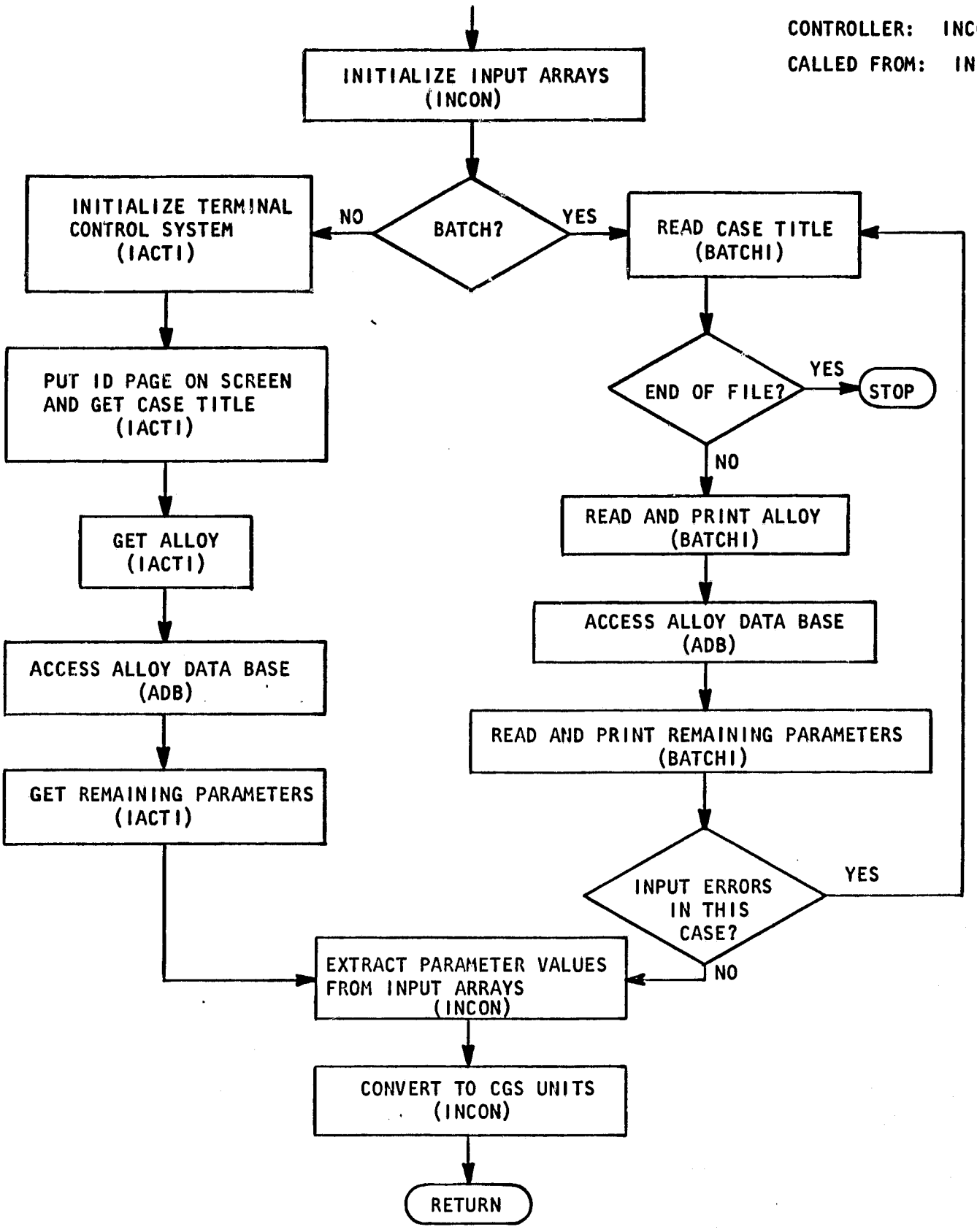
2.2 CALCULATION SECTION

CONTROLLER: Main

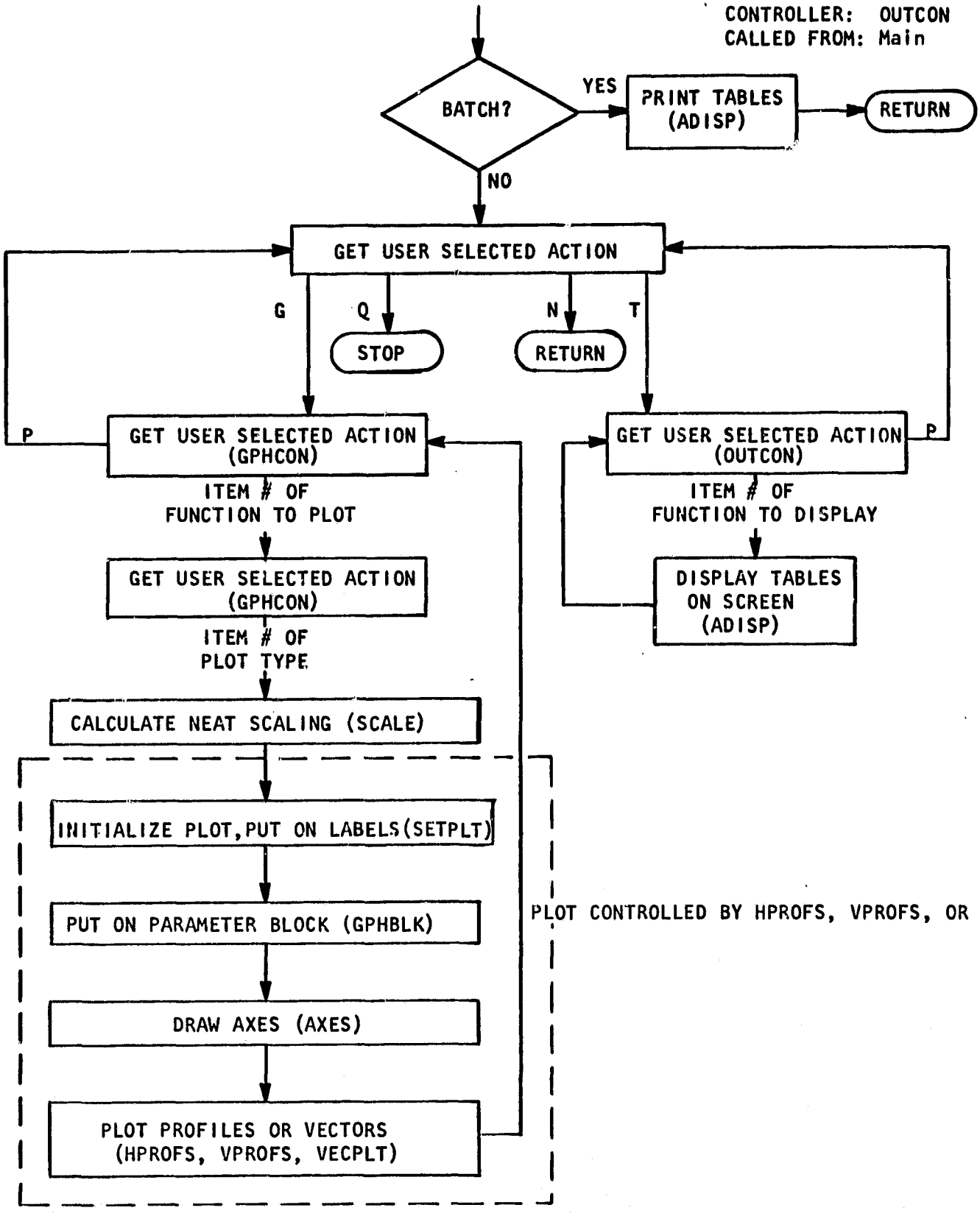


2.3 INPUT SECTION

CONTROLLER: INCON
CALLED FROM: INIT



2.4 OUTPUT SECTION



SECTION 3
MODULES

3.1 ALPHABETICAL LIST OF MODEL 1 SUBROUTINES

The list below contains only the modules that were written specifically for the solidification model. Off-the-shelf routines used by the model are listed in Sections 3.2 and 3.3. The purpose of each routine is described briefly below; details of the programming, including calling sequence descriptions, are in the program comments.

NAME	SECTION	FUNCTION
ADB	Input	Retrieve phase diagram, densities and viscosity from alloy data base.
ADISP	Output	Displays any array on the screen or in printed form.
AXES	Output	Draws and labels axes on all plots.
BATCHI	Input	Controls batch mode card input.
EST	Calculation	Calculates initial estimated solution (see Vol 1, 4.3).
FRECKL	Calculation	Sets flag if freckling condition detected (see Vol 1, 4.10).
GPBLK	Output	Puts parameter block on plot.
GPHCON	Output	Controls interactive graphics.
HPROFS	Output	Controls horizontal profile plots.
IACTI	Input	Controls interactive input.
INCON	Input	Controls input section.
INIT	Calculation	Initialization routine.
LFRAC	Calculation	Calculates g_L (see Vol 1, 4.7).
MACSEG	Calculation	Calculates $\overline{C}_s$ (see Vol 1, 4.9).
MSG	Calculation	Puts any brief message on terminal screen.
OUTCON	Output	Controls output section.
PERM	Calculation	Calculates permeability according to equation (1.3.6.10).
PSETUP	Calculation	Calculates $\vec{A}$, and B and sets up boundary conditions (see Vol 1, 4.4 and 4.5).
PSOLVE	Calculation	Solves the pressure equation (see Vol 1, 4.5)

NAME	SECTION	FUNCTION
SCALE	Output	Calculates neat scales for plots.
SETHOL	Input	Sets up long hollerith arrays.
SETPLT	Output	Initializes all plots and puts on labels.
SSICON	Calculation	Controls convergence to steady-state solution (see Vol I, 4.8).
VECPLT	Output	Controls plots of vector fields.
VLCTY	Calculation	Calculates velocity (see Vol I, 4.6).
VPROFS	Output	Controls vertical profile plots.
WAIT	Output	Waits for the operator to enter a P.

3.2 PRIME SYSTEM ROUTINES

The following routines from the Prime system library are called by the solidification model. Documentation is available in the Prime Corporation Document, "FORTRAN Programmer's Guide", PDR 3057.

NAME	CALLED BY	FUNCTION
DATE$A	BATCHI, IACTI	Returns current calendar date.
SEARCH	ADB	Used to rewind data base file because Prime FORTRAN REWIND does not work.
TIME$A	BATCHI, IACTI	Returns current wall-clock time.

3.3 TEKTRONIX ROUTINES

The following routines from the Tektronix software libraries control the terminal and provide basic plot capability. They are documented in the Tecktronix, Inc. reports "Plot-10 Terminal Control System User's Manual", Document No. 062-1474-00 and "Plot-10 Advanced Graphing II User's Manual", Document No. 062-1530-00.

NAME	SYSTEM	FUNCTION
ANMODE	TCS	Switches to alphanumeric mode for use of FORTRAN I/O.
AOUTST	TCS	Outputs alphanumeric data through Terminal Control System.

NAME	SYSTEM	FUNCTION
BELL	TCS	Rings terminal bell.
BINITT	AGS-II	Initialize plot common area.
CHECK	AGS-II	Sets parameters in plot common area.
CHRSIZ	TCS	Sets character size.
CPLOT	AGS-II	Plots a curve (Must follow call to DSPLAY).
CSIZE	TCS	Returns dimensions of a character in raster units.
DLIMX, DLIMY	AGS-II	Set the data values corresponding to the plot window: they determine the scale of the plot.
DRWABS	TCS	Generates a line from the current beam position to the raster coordinate given in the parameter list.
DSHABS	TCS	The same function as DRWABS, except the line is dashed.
DSPLAY	AGS-II	Displays axes and curve on screen.
DWINDO	TCS	Sets correspondence between user coordinates and virtual window.
GRID	AGS-II	Draws axes.
INITT	TCS	Initializes the Terminal Control System.
LINE	AGS-II	Sets a dashed line pattern for use by CPLOT.
LINHGT, LINWDT	TCS	Determine the height or length of a block of text.
MOVABS	TCS	Moves the beam to a given point.
NEWPAG	TCS	Clears the screen.
NPTS	AGS-II	Tells CPLOT how many points are on curve.
SLIMX, SLIMY	AGS-II	Sets the virtual window.
TERM	TCS	Identifies terminal hardware type to Terminal Control System.
XFRM, YFRM	AGS-II	Set form of major tic marks.

NAME	SYSTEM	FUNCTION
XLEN, YLEN	AGS-II	Set length of major tic marks.
XLOC, XLOCTP	AGS-II	Set locations of the x-axes.
XMTCS, YMTCS	AGS-II	Set the number of minor tic mark intervals.
XNEAT, YNEAT	AGS-II	Set the "neat" tic mark condition.
XTICS, YTICS	AGS-II	Set the number of major tic mark intervals.
XZERO, YZERO	AGS-II	Set the zero-suppression flag for tic mark values.
YLOC, YLOCRT	AGS-II	Set locations of the y-axes.

SECTION 4

KEY PROGRAM SYMBOLS

A description of all common block items is at the end of subroutine INIT.

PROGRAM SYMBOL	COMMON BLOCK	DEFINITION (See Volume I, Section 2)
A	/PRSSEQ/	$\overline{A}$
ALNAM	/ALLOY/	Element 1 is the hollerith solvent name. Element 2 is the hollerith solute name.
AVCS	/PHYS/	$\overline{C}_s$
B	/PRSSEQ/	B
CL	/PHYS/	C_L
CLØ	/ALLOY/	C_o
DCLDT	/ALLOY/	dC_L/dT
DCLDTM	/PHYS/	$\partial C_L/\partial t$
DGLDTM	/PHYS/	$\partial g_L/\partial t$
DRHDC	/ALLOY/	$d\rho_L/dC_L$
DRHDTM	/PHYS/	$\partial \rho_L/\partial t$
DTDTM	/PROCSS/	$\partial T/\partial t = \varepsilon$
DTDX	/PROCSS/	$\partial T/\partial x = G$
DX	/MESH/	Δx
DXM	/PROCSS/	$x_L - x_E$
DY	/MESH/	Δy
DYM	/PROCSS/	L
GAMMA	/PMBLTY/	γ'
GL	/PHYS/	g_L
GRAV	/PROCSS/	g
KK	/PHYS/	K
MAXSSI	/SSI/	M_{SSI}
MAXSOR	/SOR/	M_{SOR}

PROGRAM SYMBOL	COMMON BLOCK	DEFINITION (See Volume I, Section 2)
NI	/MESH/	N_i
NJ	/MESH/	N_j
P	/PHYS/	p
RHL	/PHYS/	ρ_L
RHLE	/ALLOY/	ρ_{LE}
RHS	/ALLOY/	ρ_S
RHSE	/ALLOY/	ρ_{SE}
T	/PHYS/	T
TE	/ALLOY/	T_E
TLO	/ALLOY/	T_L
V	/PHYS/	$\bar{V}$
VISC	/ALLOY/	μ
X, XX	/MESH/	x, $(x-x_E)/(x_L-x_E)$
Y, YY	/MESH/	y, y/L

SECTION 5
PROGRAM CONFIGURATION ON THE PRIME

The procedures described in this section are used to maintain the program on the Prime 400 system.

5.1 COMPILATION

If the FORTRAN source code is stored in a file named MPSI.PGM, then it can be compiled by entering the command

```
FTN MPSI.PGM 2/500
```

5.2 LOADING

The Prime utility for loading and running segmented programs in SEG. It can be used to build a run file as follows:

```
SEG
LO #MPSI
LO B_MPSI.PGM
LIB ¯APPLB
LIB TCS500
LIB
SAV
Q
```

5.3 EXECUTION COMMAND FILES

After the run file #MPSI has been built, the program can be executed by entering the commands described in Volume III. The execution is set up and controlled by two command files listed below:

Command File MPSI

```
OPEN ICARD 1 1
OPEN M1.D.B 3 1
CO -END
```

Command File MPSI.BATCH

```
OPEN CARDS 1 1
OPEN M1.D.B 3 1
OPEN PRINT 2 2
SEG #MPSI
C 1 2 3
CO -END
```

CARDS is a disk file containing the batch card input, and ICARD contains the single word INTERACTIVE. CARDS or ICARD is accessed by the program via FORTRAN logical unit number 5. M1.D.B is the alloy data base accessed via logical unit

number 7. PRINT is the batch printed output written on logical unit 6. No printed output is generated by an interactive mode run.

C

APPENDIX A

LISTING

```
(0001)  C
(0002)  C-----------------------------------------------------------------
(0003)  C
(0004)  C CONTROLLER FOR CALCULATION OF MACROSEGREGATION IN A CASTING INGOT
(0005)  C
(0006)  C MPS SOLIDIFICATION MODEL 1  (12/79)
(0007)  C
(0008)  C          * DEVELOPED FOR MARSHALL SPACE FLIGHT CENTER
(0009)  C            BY THE GENERAL ELECTRIC CO., HUNTSVILLE OPERATIONS
(0010)  C            OF THE SPACE DIVISION.
(0011)  C
(0012)  C          * A DESCRIPTION OF THE MODEL IS IN THE GE DOCUMENT
(0013)  C
(0014)  C            MPS SOLIDIFICATION MODEL -
(0015)  C                 VOLUME I:   FORMULATION AND ANALYSIS
(0016)  C                 VOLUME II:  SOFTWARE DOCUMENTATION
(0017)  C                 VOLUME III: OPERATING MANUAL
(0018)  C
(0019)  C-----------------------------------------------------------------
(0020)  C
(0021)        LOGICAL AGAIN, FIRST, STABLE
(0022)  C
(0023)  C INITIALIZE PROGRAM / GO TO NEXT CASE
(0024)  C
(0025)        FIRST= .TRUE.
(0026)   200  CALL INIT (FIRST)
(0027)  C
(0028)  C ITERATE TO THE SOLUTION OF THE NONLINEAR STEADY STATE SYSTEM.
(0029)  C SOLVE FOR PRESSURE, VELOCITY AND FRACTION LIQUID.
(0030)  C EST CALCULATES THE INITIAL ESTIMATED SOLUTION.
(0031)  C ITERATION CONTROL PROVIDED BY SUBROUTINE SSICON.
(0032)  C
(0033)        STABLE= .TRUE.
(0034)        CALL EST
(0035)        CALL SSICON ( 1, AGAIN )
(0036)   500  CONTINUE
(0037)  C
(0038)        CALL PSETUP
(0039)        CALL PSOLVE
(0040)  C
(0041)        CALL VLCTY
(0042)        CALL FRECKL (STABLE)
(0043)  C
(0044)        CALL LFRAC
(0045)  C
(0046)        CALL SSICON ( 2, AGAIN )
(0047)        IF (AGAIN.AND.STABLE) GO TO 500
(0048)  C
(0049)  C CALCULATE THE MACROSEGREGATION
(0050)  C
(0051)        CALL MACSEG
(0052)  C
(0053)  C OUTPUT PRINTED TABLES AND PLOTS
(0054)  C
(0055)        CALL OUTCON (FIRST)
(0056)  C
(0057)        FIRST= .FALSE.
(0058)        GO TO 200
(0059)  C
(0060)        END
```

```
(0061)        SUBROUTINE INIT (FIRST)
(0062)  C
(0063)  C PROGRAM INITIALIZATION, NEXT CASE LOGIC
(0064)  C
(0065)  C-----------------------------------------------------------------------
(0066)  C     FIRST   (INPUT)     TRUE DURING FIRST CASE.
(0067)                            LOGICAL FIRST
(0068)  C-----------------------------------------------------------------------
(0068)  C COMMON BLOCKS                              (SEE INIT FOR DOCUMENTATIO
(0068)  C
(0068)        COMMON /MESH/    X(50), XX(50), DX, IDIM, NI, NIM, NIP,
(0068)       *                 Y(50), YY(50), DY, JDIM, NJ, NJM, NJP
(0068)  C
(0068)        COMMON /PHYS/    T(50,50), CL(50,50), DCLDTM,
(0068)       *                 RHL(50,50), DRHDTM, GL(50,50), DGLDTM(50,50),
(0068)       *                 P(52,52), V(2,50,50), AVCS(50),
(0068)       *                 KK(50,50)
(0068)                         REAL KK
(0068)  C
(0068)        COMMON /PROCSS/  DXM, DYM,
(0068)       *                 GRAV, GFORCE,
(0068)       *                 DTDTM, DTDX
(0068)  C
(0068)        COMMON /ALLOY/   ALNAM(2), ALREF(20,5), ADBREF(20),
(0068)       *                 TLO, CLO, TE, CE, DCLDT, EPR,
(0068)       *                 RHLO, DRHDC, RHLE, RHSE, RHS,
(0068)       *                 VISC
(0068)                         INTEGER ALNAM, ALREF, ADBREF
(0068)  C
(0068)        COMMON /PHBLTY/  GAMMA
(0068)  C
(0068)        COMMON /PRSSEQ/  A(2,52,52), E(52,52),
(0068)       *                 DPDY0(50), DPDYL(50), DPDXE(50)
(0068)  C
(0068)        COMMON /SOR/     XCF1, YCF1, XCF2, YCF2, CCF,
(0068)       *                 PIT, MAXSOR, EPSSOR, ESORI
(0068)                         INTEGER PIT
(0068)  C
(0068)        COMMON /SSI/     MAXSSI, EPSSSI
(0068)  C
(0068)        COMMON /CNTRL/   MODEL, BATCH,
(0068)       *                 PRCSN, NCW, RINF
(0068)                         LOGICAL BATCH
(0068)  C
(0068)        COMMON /SCR/     SCR1(2,50,50), SCR2(100)
(0068)  C
(0068)        COMMON /IO/      CARD, PRNT, CRT, ADBU,
(0068)       *                 TITLE(20,2),
(0068)       *                 TYPE(25), LN(10,25), NLN, LU(5,25), NLU,
(0068)       *                 PGHDR(20,5), KEYS(25,5), RVLU(3,25),
(0068)       *                 IFMT1(10), RFMT1(10), FMT2(5),
(0068)       *                 IFMT4(16), RFMT4(20), FMT5(7),
(0068)       *                 FMT80(3), FMTA4(2), TIFMT(16),
(0068)       *                 NLSCRN
(0068)                         INTEGER CARD, PRNT, CRT, ADBU
(0068)                         INTEGER TITLE, TYPE, PGHDR
(0068)                         DIMENSION IVLU(3,25)
(0068)                         EQUIVALENCE (RVLU,IVLU)
(0068)                         INTEGER RFMT1, FMT2, RFMT4, FMT5
(0068)                         INTEGER FMT80, FMTA4, TIFMT
```

```
      SUBROUTINE INIT (FIRST)

(0068)  C
(0068)        COMMON /GRAPH/  HSCRN, VSCRN, HW, VW,
(0068)       *                HWL, HWR, VWL, VWU,
(0068)       *                HCH, VCH,
(0068)       *                PBLK(5,10), NPBL, NPBCH
(0068)                        INTEGER HSCRN, VSCRN, HW, VW
(0068)                        INTEGER HWL, HWR, VWL, VWU
(0068)                        INTEGER HCH, VCH
(0068)                        INTEGER PBLK
(0068)  C-----------------------------------------------------------------------
(0069)  C
(0070)        IF (.NOT.FIRST) GO TO 100
(0071)  C
(0072)  C INITIALIZE FIXED COMMON PARAMETERS
(0073)  C
(0074)  C * /MESH/
(0075)        IDIM= 50
(0076)        JDIM= 50
(0077)  C * /CNTRL/
(0078)        MODEL= 1
(0079)        PRCSN= 1.E-6
(0080)        NCW= 4
(0081)        RINF= 1.E37
(0082)        NMSG= 20
(0083)  C * /IO/
(0084)        CARD= 5
(0085)        PRNT= 6
(0086)        CRT= 1
(0087)        ADBU= 7
(0088)        NLSCRN= 64
(0089)  C * /GRAPH/
(0090)        HSCRN= 1024
(0091)        VSCRN= 781
(0092)        HW= 734
(0093)        VW= 575
(0094)  C
(0095)  C TEST FOR BATCH OR INTERACTIVE MODE
(0096)  C
(0097)        READ (CARD,50) MODE
(0098)   50   FORMAT (A4)
(0099)        BATCH= (MODE .NE. 4HINTE)
(0100)  C
(0101)  C GET INPUT
(0102)  C
(0103)   100  CALL INCON (FIRST)
(0104)  C
(0105)  C ASSIMILATE INPUT DATA
(0106)  C
(0107)  C * MESH
(0108)        NIM= NI-1
(0109)        NIP= NI+1
(0110)        DX= DXM/FLOAT(NIM)
(0111)        DO 200 I=1,NI
(0112)        X(I)= FLOAT(I-1)* DX
(0113)        XX(I)= (X(I)-X(1))/ DXM
(0114)   200  CONTINUE
(0115)        NJM= NJ-1
(0116)        NJP= NJ+1
(0117)        DY= DYM/FLOAT(NJM)
(0118)        DO 250 J=1,NJ
```

```
      SUBROUTINE INIT (FIRST)

(0119)        Y(J)= FLOAT(J-1)* DY
(0120)        YY(J)= (Y(J)-Y(1))/ DYM
(0121)   250  CONTINUE
(0122)  C * TEMPERATURE, DENSITY, LIQUID COMPOSITION
(0123)        TLO= TE + (CLO-CE)/DCLDT
(0124)        DTDX= (TLO-TE)/DXM
(0125)        DCLDTM= DCLDT * DTDTM
(0126)        RHLC= RHLE + DRHDC* (CLO-CE)
(0127)        DRHDTM= DRHDC * DCLDTM
(0128)        DO 300 J=1,NJ
(0129)        DO 300 I=1,NI
(0130)        T(I,J)= TE + DTDX*( X(I)-X(1) )
(0131)        CL(I,J)= CE + DCLDT*( T(I,J)-T(1,J) )
(0132)        RHL(I,J)= RHLE + DRHDC*( CL(I,J)-CL(1,J) )
(0133)   300  CONTINUE
(0134)  C * SOR MESH-DEPENDENT COEFFICIENTS
(0135)        CCF= .5* (DX**2) * (DY**2) /(DX**2 + DY**2)
(0136)        XCF1= .5* CCF/ DX
(0137)        XCF2= CCF/ (DX**2)
(0138)        YCF1= .5* CCF/ DY
(0139)        YCF2= CCF/ (DY**2)
(0140)  C
(0141)        RETURN
(0142)  C-----------------------------------------------------------------
(0143)  C COMMON DOCUMENTATION
(0144)  C /MESH/    X          HORIZONTAL MESH POINTS (CM) - DISTANCE FROM XF.
(0145)  C           XX         NORMALIZED MESH POINTS - (X-XE)/(XL-XE).
(0146)  C           DX         HORIZONTAL MESH SIZE (CM).
(0147)  C           IDIM       X DIMENSION OF ALL ARRAYS.
(0148)  C           NI         NUMBER OF X MESH POINTS IN CURRENT CALCULATION.
(0149)  C           NIM,NIP    NI-1, NI+1
(0150)  C           Y          VERTICAL MESH POINTS (CM) - PHYSICAL DISTANCE
(0151)  C                      FROM BOTTOM OF MUSHY ZONE.
(0152)  C           YY         NORMALIZED MESH POINTS - Y/L.
(0153)  C           DY         VERTICAL MESH SIZE (CM).
(0154)  C           JDIM       Y DIMENSION OF ALL ARRAYS.
(0155)  C           NJ         NUMBER OF Y MESH POINTS IN CURRENT CALCULATION.
(0156)  C           NJM,NJP    NJ-1, NJ+1
(0157)  C /PHYS/    T          TEMPERATURE FIELD (DEG C).
(0158)  C           CL         LIQUID COMPOSITION (WT PCT SOLUTE).
(0159)  C           DCLDTM     PARTIAL OF CL WRT TIME.
(0160)  C           RHL        LIQUID DENSITY (GM/CM**3).
(0161)  C           DRHDTM     PARTIAL OF RHL WRT TIME.
(0162)  C           GL         VOLUME FRACTION LIQUID.
(0163)  C           DGLDTM     PARTIAL OF GL WRT TIME.
(0164)  C           P          MODIFIED PRESSURE (DYNES/CM**2).
(0165)  C           V          VELOCITY (CM/S).
(0166)  C           AVCS       FINAL LOCAL AVERAGE COMPOSITION (WT PCT SOLUTE).
(0167)  C           KK         PERMEABILITY (CM**2)
(0168)  C /PROCSS/  DXM        WIDTH OF MUSHY ZONE (CM).
(0169)  C           DYM        HEIGHT OF MUSHY ZONE (CM).
(0170)  C           GRAV       ACCELERATION DUE TO GRAVITY IN THE -Y DIRECTION.
(0171)  C                      (CM/S**2), POSITIVE DOWNWARD.
(0172)  C           GFORCE     GRAV IN UNITS OF G=980.66 CM/S**2.
(0173)  C                      USED ONLY FOR I/C.
(0174)  C           DTDTM      PARTIAL OF TEMPERATURE WRT TIME (DEG C/S).
(0175)  C           DTDX       PARTIAL OF TEMPERATURE WRT X (DEG C/CM).
(0176)  C /ALLOY/   ALNAM      ALLOY NAME (ALPHANUMERIC).
(0177)  C           ALREF      SOURCES OF ALLOY DATA (ALPHANUMERIC).
(0178)  C           ADBREF     ALLOY DATA BASE IDENTIFIER (ALPHANUMERIC).
```

```
(0179)  C           TLO       LIQUIDUS TEMPERATURE AT COMPOSITION CLO (DEG C).
(0180)  C           CLO       BULK LIQUID COMPOSITION (WT PCT SOLUTE).
(0181)  C           TE        EUTECTIC TEMPERATURE (DEG C).
(0182)  C           CE        EUTECTIC COMPOSITION (WT PCT SOLUTE).
(0183)  C           DCLDT     SLOPE OF PHASE DIAGRAM LIDQUIDUS (PCT SOLUTE/DEG C)
(0184)  C           EPR       EQUILIBRIUM PARTITION RATIO.
(0185)  C           RHLO      DENSITY OF BULK LIQUID (GM/CM**3).
(0186)  C           DRHDC     PARTIAL OF RHL WRT CL (GM/CM**3 / PCT SOLUTE).
(0187)  C           RHLE      LIQUID EUTECTIC DENSITY (GM/CM**3).
(0188)  C           RHSE      SOLID EUTECTIC DENSITY (GM/CM**3).
(0189)  C           RHS       SOLID ALLOY DENSITY (GM/CM**3).
(0190)  C           VISC      VISCOSITY OF LIQUID (GM/(CM*S)).
(0191)  C /PMBLTY/  GAMMA     PERMEABILITY COEFFICIENT (CM**2).
(0192)  C /PRSSEQ/  A         COEFFICIENT OF GRAD(P) IN PRESSURE EQUATION.
(0193)  C           B         CONSTANT TERM IN PRESSURE EQUATION.
(0194)  C           DPDYO     PARTIAL OF PRESSURE WRT Y AT BOTTOM OF INGOT.
(0195)  C           DPDYL     PARTAL OF PRESSURE WRT Y AT TOP OF INGOT.
(0196)  C           DPDXE     PARTIAL OF P WRT X AT EUTECTIC ISOTHERM.
(0197)  C /SOR/     XCF1,XCF2  COEFFS OF-X DERIVATIVE TERMS IN PRESSURE EQN.
(0198)  C           YCF1,YCF2  COEFFS OF Y DERIVATIVE TERMS IN PRESSURE EQN.
(0199)  C           CCF       COEFFICIENT OF CONSTANT TERM IN PRESSURE EQN.
(0200)  C           PIT       AFTER CALL TO PSOLVE, NUMBER OF PRESSURE
(0201)  C                     ITERATIONS USED.
(0202)  C           MAXSOR    MAXIMUM NUMBER OF ITERATIONS ON SOLUTION OF
(0203)  C                     PRESSURE EQUATION (SOR TECHNIQUE).
(0204)  C           EPSSOR    MINIMUM CONVERGENCE CRITERION FOR SOR TECHNIQUE.
(0205)  C           ESORI     SOR CONVERGENCE CRITERION DURING CURRENT STEADY
(0206)  C                     STATE ITERATION. SET IN SSICON.
(0207)  C /SSI/     MAXSSI    MAXIMUM NUMBER OF ITERATIONS ON STEADY
(0208)  C                     STATE SOLUTION.
(0209)  C           EPSSSI    CONVERGENCE CRITERION FOR STEADY STATE SOLUTION.
(0210)  C /CNTRL/   MODEL     MODEL NUMBER FOR IDENTIFICATION PURPOSES.
(0211)  C           BATCH     TRUE FOR BATCH MODE.
(0212)  C                     FALSE FOR INTERACTIVE MODE.
(0213)  C           PRCSN     1.E-N WHERE N IS THE PRECISION OF THE MACHINE.
(0214)  C           NCW       NUMBER OF CHARACTERS THAT CAN BE STORED IN AN
(0215)  C                     INTEGER. (DEFAULT LENGTH)
(0216)  C           RINF      LARGEST FLOATING POINT NUMBER IN MACHINE.
(0217)  C /SCR/     SCR1      SCRATCH AREA 1.
(0218)  C           SCR2      SCRATCH AREA 2.
(0219)  C /IO/      CARD      LOGICAL UNIT NUMBER FOR CARD INPUT.
(0220)  C           PRNT      LOGICAL UNIT NUMBER FOR PRINTED OUTPUT.
(0221)  C           CRT       LOGICAL UNIT NUMBER OF CRT FOR INTERACTIVE
(0222)  C                     I/O AND GRAPHICAL OUTPUT.
(0223)  C           ADBU      LOGICAL UNIT NUMBER OF INPUT ALLOY DATA BASE.
(0224)  C           TITLE     TWO LINE ALPHANUMERIC TITLE USED ON ALL OUTPUT.
(0225)  C           TYPE      TYPE(K) IS THE TYPE OF THE KTH INPUT VARIABLE.
(0226)  C                     1  INTEGER
(0227)  C                     2  REAL
(0228)  C           LN,NLN    LN(1-NLN,K) IS THE ALPHANUMERIC NAME OF THE KTH
(0229)  C                     INPUT VARIABLE.
(0230)  C           LU,NLU    LU(1-NLU,K) IS THE ALPHANUMERIC UNITS DESCRIPTION
(0231)  C                     OF THE KTH INPUT VARIBLE.
(0232)  C           PGHDR     PGHDR(1-20,J) IS THE ALPHANUMERIC HEADER FOR THE
(0233)  C                     JTH PAGE (INTERACTIVE) OR PARAGRAPH (BATCH) OF INPU
(0234)  C           KEYS      KEYS(I,J) IS THE INDEX IN TYPE,LN,LU,RVLU OF THE
(0235)  C                     ITH VARIABLE IN THE JTH PAGE/PARAGRAPH OF INPUT.
(0236)  C           RVLU      RVLU(1,2,3,K) ARE THE VALUE, LOWER BOUND, AND
(0237)  C                     UPPER BOUND, RESPECTIVELY, OF THE KTH INPUT VARIABL
(0238)  C           FMT S     THE ARRAYS WITH FMT IN THEIR NAMES ARE FORMAT
```

```
      SUBROUTINE INIT (FIRST)


(0239)  C                       STATEMENTS SET UP IN INCON.
(0240)  C             NLSCRN    MAXIMUM NUMBER OF LINES THAT FIT ON THE CRT SCREEN.
(0241)  C /GRAPH/     HSCRN     WIDTH OF CRT SCREEN IN RASTER UNITS.
(0242)  C             VSRCN     HEIGHT OF CRT SCREEN IN RASTER UNITS.
(0243)  C             HW,VW     HORIZONTAL,VERTICAL RASTER LENGTH OF PLOT WINDOW.
(0244)  C             HWL,HWR   HORIZONTAL RASTER COORDINATE OR LEFT,RIGHT SIDE
(0245)  C                       OF PLOT WINDOW.
(0246)  C             VWL,VWU   VERTICAL RASTER COORDINATE OF LOWER,UPPER SIDE
(0247)  C                       OF PLOT WINDOW.
(0248)  C             PBLK      PARAMETER BLOCK FOR PLOTS.
(0249)  C             NPBL      NUMBER OF LINES IN PARAMETER BLOCK.
(0250)  C             NPRCH     NUMBER OF CHARACTERS PER LINE IN PARAMETER BLOCK.
(0251)  C-----------------------------------------------------------------------
(0252)        END
PROGRAM SIZE:   PROCEDURE - 000631   LINKAGE - 000217   STACK - 000030
0000 ERRORS [<INIT  >FTN-REV15.3]
```

SUBROUTINE EST

```
(0253)        SUBROUTINE EST
(0254)  C
(0255)  C INITIAL ESTIMATED PRESSURE, GL AND DGLDTM.
(0256)  C EXACT SOLUTION FOR CASE WITH PLANAR ISOTHERMS,
(0257)  C LINEAR VARIATION OF RHL, CL AND T IN THE MUSHY ZONE,
(0258)  C AND NO GRAVITATIONAL FORCE.
(0259)  C
(0260)  C-----------------------------------------------------------------------
(0260)  C COMMON BLOCKS                    (SEE INIT FOR DOCUMENTATION
(0260)  C-----------------------------------------------------------------------
(0261)  C
(0262)  C GL, DGLDTM, VELOCITIES.
(0263)  C
(0264)        E= -1./(1.-EPR)
(0265)        GLC= (RHS/RHSE -1.) * CE**E
(0266)        GLD= CL0**E + GLC
(0267)        GE= (CE**E + GLC)/GLD
(0268)  C
(0269)        DO 200 I=1,NI
(0270)        DO 200 J=1,NJ
(0271)        GL(I,J)= (CL(I,J)**E + GLC)/GLD
(0272)        DGLDTM(I,J)= E* (GL(I,J)+ (RHSE/RHS-1.)*GE )* DCLDTM/ CL(I,J)
(0273)        V(1,I,J)= ( (GL(I,J)*(RHS-RHL(I,J))+GE*(RHSE-RHS))/
(0274)       *            (RHL(I,J)*GL(I,J)) )* DTDTM/DTDX
(0275)        V(2,I,J)= 0.
(0276)   200  CONTINUE
(0277)  C
(0278)  C PERMEABILITY
(0279)  C
(0280)        CALL PERM
(0281)  C
(0282)  C ESTIMATED PRESSURE FIELD. ( P= P0- RHL*GRAV*(L-Y) )
(0283)  C
(0284)        DO 300 J=1,NJ
(0285)        P(NI+1,J+1)= 0.
(0286)        FI= ( -VISC*GL(NI,J)/KK(NI,J) )* V(1,NI,J)
(0287)        DO 300 II=2,NI
(0288)        I= NI-II+1
(0289)        FIP= FI
(0290)        FI= ( -VISC*GL(I,J)/KK(I,J) )* V(1,I,J)
(0291)        P(I+1,J+1)= P(I+2,J+1)- .5*(FI+FIP)*DX
(0292)   300  CONTINUE
(0293)  C
(0294)        RETURN
(0295)        END
PROGRAM SIZE:   PROCEDURE - 000456    LINKAGE - 000120    STACK - 000022
0000 ERRORS [<EST   >FTN-REV15.3]
```

```
      SUBROUTINE FRECKL (STABLE)

(0296)        SUBROUTINE FRECKL (STABLE)
(0297)  C
(0298)  C CHECKS FOR THE TYPE OF INSTABILITY THAT CAUSES THE FORMATION
(0299)  C OF FRECKLES OR A-SEGREGATES.
(0300)  C
(0301)  C THE INSTABILITY OCCURS WHEN THE FLUID VELOCITY IS GREATER THAN
(0302)  C THE ISOTHERM VELOCITY SO THAT DENDRITE REMELTING OCCURS.
(0303)  C
(0304)  C     D(T)/D(TIME) = PARTL(T)/PARTL(TIME) + V*GRAD(T) .GT. 0
(0305)  C
(0306)  C-----------------------------------------------------------------
(0307)  C     STABLE    (OUTPUT)  TRUE IF NO INSTABILITY DETECTED.
(0308)                            LOGICAL STABLE
(0309)  C-----------------------------------------------------------------
(0309)  C COMMON BLOCKS                          (SEE INIT FOR DOCUMENTATION
(0309)  C-----------------------------------------------------------------
(0310)  C
(0311)        STABLE= .TRUE.
(0312)  C
(0313)        DO 200 J=1,NJ
(0314)        DO 200 I=1,NI
(0315)        STABLE= STABLE .AND.
(0316)       *    .NOT.( DTDTM*V(1,I,J)*DTLX .GT. 0. )
(0317)   200  CONTINUE
(0318)  C
(0319)        IF (.NOT.STABLE) CALL MSG( 80, 'OHCALCULATION ABORTED - FLOW INST
(0320)       *BILITY LEADING TO FRECKLE FORMATION DETECTED.     )
(0321)  C
(0322)        RETURN
(0323)        END
PROGRAM SIZE:   PROCEDURE - 000154    LINKAGE - 000040    STACK - 000016
0000 ERRORS [<FRECKL>FTN-REV15.3]
```

SUBROUTINE LFRAC

```
(0324)          SUBROUTINE LFRAC
(0325)    C
(0326)    C CALCULATES LOCAL VOLUME FRACTION LIQUID.
(0327)    C INTEGRATE LOCAL SOLUTE REDISTRIBUTION EQUATION FROM LIQUIDUS
(0328)    C TO X(I) ALONG CONSTANT Y.  ASSUMES STEADY STATE SOLUTION.
(0329)    C
(0330)    C-----------------------------------------------------------------
(0330)    C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0330)    C-----------------------------------------------------------------
(0331)    C
(0332)    C CALCULATE FRACTION LIQUID AND DGLDTM.
(0333)    C
(0334)          E= -1./ (1.-EPR)
(0335)    C
(0336)          DO 200 J=1,NJ
(0337)          EI= E* ( RHL(NI,J)/RHS )* ( 1. + V(1,NI,J)*DTDX/DTDTM )
(0338)          GL(NI,J)= 1.
(0339)          DGLDTM(NI,J)= EI* (GL(NI,J)/CL(NI,J))* DCLDTM
(0340)    C
(0341)          DO 200 II=2,NI
(0342)          I= NI-II+1
(0343)          EI1= EI
(0344)          EI= E* ( RHL(I,J)/RHS  )* ( 1. + V(1,I,J)*DTDX/DTDTM )
(0345)          GL(I,J)= GL(I+1,J)* ( CL(I,J)/CL(I+1,J) )**( .5*(EI+EI1) )
(0346)          DGLDTM(I,J)= EI* (GL(I,J)/CL(I,J))* DCLDTM
(0347)     200  CONTINUE
(0348)    C
(0349)    C CALCULATE NEW PERMEABILITIES.
(0350)    C
(0351)          CALL PERM
(0352)    C
(0353)          RETURN
(0354)          END
PROGRAM SIZE:   PROCEDURE - 000252    LINKAGE - 000074    STACK - 000020
0000 ERRORS [<LFRAC >FTN-REV15.3]
```

SUBROUTINE MACSEG

```
(0355)        SUBROUTINE MACSEG
(0356)  C
(0357)  C CALCULATE THE LOCAL AVERAGE COMPOSITION OF THE FINAL SOLID.
(0358)  C STEADY STATE FORMULATION.
(0359)  C
(0360)  C-------------------------------------------------------------------
(0360)  C COMMON BLOCKS                    (SEE INIT FOR DOCUMENTATION
(0360)  C-------------------------------------------------------------------
(0361)  C
(0362)        DO 300 J=1,NJ
(0363)  C
(0364)  C INTEGRATE SOLID INTERFACE COMPOSITION OVER DENDRITIC GROWTH.
(0365)  C
(0366)        CSI= 0.
(0367)  C
(0368)        DO 200 I=2,NI
(0369)        CSI= CSI + EPR*.5*(GL(I,J)*CL(I,J)+GL(I-1,J)*CL(I-1,J))*
(0370)       *              (ALOG(GL(I,J))-ALOG(GL(I-1,J)))
(0371)   200  CONTINUE
(0372)  C
(0373)  C INCORPORATE EUTECTIC COMPOSITION
(0374)  C
(0375)        AVCS(J)= (RHS*CSI + RHSE*GL(1,J)*CE) /
(0376)       *         (RHS*(1.-GL(1,J)) + RHSE*GL(1,J))
(0377)   300  CONTINUE
(0378)  C
(0379)        RETURN
(0380)        END
PROGRAM SIZE:   PROCEDURE - 000221    LINKAGE - 000060    STACK - 000024
0000 ERRORS [<MACSEG>FTN-REV15.3]
```

SUBROUTINE PERM

```
(0381)          SUBROUTINE PERM
(0382)    C
(0383)    C PERMEABILITY AND POROSITY MODELS
(0384)    C
(0385)    C-------------------------------------------------------------------
(0385)    C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0385)    C-------------------------------------------------------------------
(0386)    C
(0387)    C ISOTROPIC MODEL
(0388)    C
(0389)          DO 200 J=1,NJ
(0390)          DO 200 I=1,NI
(0391)          KK(I,J)= AMIN1( GAMMA* GL(I,J)**2,.GAMMA )
(0392)     200  CONTINUE
(0393)    C
(0394)          RETURN
(0395)          END
PROGRAM SIZE:   PROCEDURE - 000067    LINKAGE - 000040    STACK - 000014
0000 ERRORS [<PERM  >FTN-REV15.3]
```

SUBROUTINE PSETUP

```
(0396)        SUBROUTINE PSETUP
(0397)  C
(0398)  C CALCULATES COEFFICIENTS OF PRESSURE EQUATON.
(0399)  C SETS UP BOUNDARY CONDITIONS THAT DEPEND ON SOLUTION VARIABLES.
(0400)  C
(0401)  C-----------------------------------------------------------------
(0401)  C COMMON BLOCKS                          (SEE INIT FOR DOCUMENTATIO
(0401)  C-----------------------------------------------------------------
(0402)  C STATEMENT FUNCTIONS
(0403)  C
(0404)        SF(IP,JP)=  KK(IP,JP)* RHL(IP,JP)/ VISC
(0405)  C
(0406)        ASF(IP,JP)= ALOG( SF(IP,JP) )
(0407)  C
(0408)        BSF(IP,JP)= ALOG( SF(IP,JP)* RHL(IP,JP) ).
(0409)  C
(0410)  C-----------------------------------------------------------------
(0411)  C
(0412)  C X-DERIVATIVE CONTRIBUTIONS
(0413)  C
(0414)        DO 250 J=1,NJ
(0415)        A(1,2,  J+1)= ( ASF(2,J)-ASF(1,J) )/ DX
(0416)  C
(0417)        DO 200 I=2,NIM
(0418)        A(1,I+1,J+1)= ( ASF(I+1,J)-ASF(I-1,J) )/ (2.*DX)
(0419)   200  CONTINUE
(0420)  C
(0421)        A(1,NIP,J+1)= ( ASF(NI,J)-ASF(NI-1,J) )/ DX
(0422)   250  CONTINUE
(0423)  C
(0424)  C Y-DERIVATIVE CONTRIBUTIONS AND REMAINING TERMS
(0425)  C
(0426)        DO 350 I=1,NI
(0427)        A(2,I+1,2  )= ( -ASF(I,3)+4.*ASF(I,2)-3.*ASF(I,1) )/ (2.*DY)
(0428)  C
(0429)        B(I+1,2  )= GRAV* RHL(I,1)*
(0430)       *                ( -BSF(I,3)+4.*BSF(I,2)-3.*BSF(I,1) )/(2.*DY) -
(0431)       *             ( 1./SF(I,1) )*
(0432)       *                ( (RHL(I,1)-RHS)*DGLDTM(I,1) + GL(I,1)*DRHDTM )
(0433)        B(I+1,2)= B(I+1,2) -RHL(NI,1)*GRAV*A(2,I+1,2)
(0434)  C
(0435)        DO 300 J=2,NJM
(0436)        A(2,I+1,J+1)= ( ASF(I,J+1)-ASF(I,J-1) )/(2.*DY)
(0437)  C
(0438)        B(I+1,J+1)= GRAV* RHL(I,1)*
(0439)       *                ( BSF(I,J+1)-BSF(I,J-1) )/(2.*DY) -
(0440)       *             ( 1./SF(I,J) )*
(0441)       *                ( (RHL(I,J)-RHS)*DGLDTM(I,J) + GL(I,J)*DRHDTM )
(0442)        B(I+1,J+1)= B(I+1,J+1) -RHL(NI,J)*GRAV*A(2,I+1,J+1)
(0443)   300  CONTINUE
(0444)  C
(0445)        A(2,I+1,NJP)= ( 3.*ASF(I,NJ)-4.*ASF(I,NJ-1)+ASF(I,NJ-2) )/ (2.*D
(0446)  C
(0447)        B(I+1,NJP)= GRAV* RHL(I,NJ)*
(0448)       *                ( 3.*BSF(I,NJ)-4.*BSF(I,NJ-1)+BSF(I,NJ-2) )/(2.*DY)
(0449)       *             ( 1./SF(I,NJ) )*
(0450)       *                ( (RHL(I,NJ)-RHS)*DGLDTM(I,NJ) + GL(I,NJ)*DRHDTM )
(0451)        B(I+1,NJP)= B(I+1,NJP) -RHL(NI,NJ)*GRAV*A(2,I+1,NJP)
(0452)   350  CONTINUE
(0453)  C
```

```
      SUBROUTINE PSETUP

(0454)  C EUTECTIC SHRINKAGE BOUNDARY CONDITION
(0455)  C
(0456)        DO 500 J=1,NJ
(0457)        DPDXE(J)= -(RHSF-RHL(1,J))/RHL(1,J)* (-DTDTM/DTDX) *
(0458)       *          (-VISC* GL(1,J)/ KK(1,J) )
(0459)   500  CONTINUE
(0460)  C
(0461)  C BOUNDARY CONDITIONS AT TOP AND BOTTOM
(0462)  C
(0463)        DO 600 I=1,NI
(0464)        DPDYO(I)= (RHL(NI,1 )-RHL(I,1 ))* GRAV
(0465)        DPDYL(I)= (RHL(NI,NJ)-RHL(I,NJ))* GRAV
(0466)   600  CONTINUE
(0467)  C
(0468)        RETURN
(0469)  C-----------------------------------------------------------------
(0470)  C PROGRAMMING NOTES
(0471)  C * A AND B ARE STORED ON THE EXTENDED MESH.  SEE NOTES IN
(0472)  C   SUBROUTINE PSOLVE.
(0473)  C
(0474)        END
PROGRAM SIZE:   PROCEDURE - 002021    LINKAGE - 000140    STACK - 000060
0000 ERRORS [<PSETUP>FTN-REV15.3]
```

SUBROUTINE PSOLVE

```
(0475)        SUBROUTINE PSOLVE
(0476)  C
(0477)  C SOLVES 2-D LINEAR ELLIPTIC EQUATION OF FORM (D**2)P + A*(D)P + B =.0
(0478)  C SUCCESSIVE OVERRELAXATION VERSION
(0479)  C
(0480)  C-------------------------------------------------------------------
(0480)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0480)  C-------------------------------------------------------------------
(0481)  C LOCAL DECLARATIONS
(0482)        LOGICAL OCONV
(0483)        REAL LBMAX
(0484)  C-------------------------------------------------------------------
(0485)  C
(0486)        PIT= 0
(0487)        PCUT= 1.E-3* ABS( P(2,NJP) )
(0488)        NCHK= 12
(0489)        OMEGA= 1.375
(0490)        OCONV= .FALSE.
(0491)        OPTO= 0.
(0492)  C
(0493)   200  PIT= PIT+1
(0494)        TEST= 0.
(0495)        PDELP= DELP
(0496)        DELP= 0.
(0497)  C
(0498)  C UPDATE NEUMANN CONDITIONS
(0499)  C
(0500)        DO 220 I=1,NI
(0501)        P(I+1,NJ+2)=  P(I+1,NJ) + 2.*DY* DPDYL(I)
(0502)        P(I+1,1)=     P(I+1,3)  - 2.*DY* DPDYO(I)
(0503)   220  CONTINUE
(0504)        DO 240 J=1,NJ
(0505)        P(1,J+1)= P(3,J+1) - 2.*DX* DPDXE(J)
(0506)   240  CONTINUE
(0507)  C
(0508)  C CALCULATE NEW P
(0509)  C
(0510)        DO 300 J=2,NJP
(0511)        DO 300 I=2,NI
(0512)        OLDP= P(I,J)
(0513)        P(I,J)= (1.-OMEGA)* P(I,J)
(0514)       *       +     OMEGA* ( (XCF2+XCF1*A(1,I,J))* P(I+1,J)+
(0515)       *                    (XCF2-XCF1*A(1,I,J))* P(I-1,J)+
(0516)       *                    (YCF2+YCF1*A(2,I,J))* P(I,J+1)+
(0517)       *                    (YCF2-YCF1*A(2,I,J))* P(I,J-1)+
(0518)       *                    CCF*B(I,J) )
(0519)        IF ( (ABS(P(I,J)) .GT. PCUT) .AND. (ABS(OLDP) .GT. PCUT) .AND.
(0520)       *     (P(I,J)*OLDP .GT. 0.) )
(0521)       *TEST= AMAX1 (TEST,
(0522)       *             2.* ABS(P(I,J)-OLDP)/ (ABS(P(I,J))+ABS(OLDP)) )
(0523)        DELP= DELP + ABS( P(I,J)-OLDP )
(0524)   300  CONTINUE
(0525)  C
(0526)  C DETERMINE OPTIMAL OMEGA
(0527)  C
(0528)        IF (OCONV) GO TO 400
(0529)        IF ( MOD(PIT,NCHK).NE.0 ) GO TO 400
(0530)        LBMAX= DELP/ PDELP
(0531)        IF ( LBMAX .LT. .99 ) GO TO 350
(0532)        IF ( PIT. EQ. NCHK ) GO TO 400
```

SUBROUTINE PSOLVE

```
(0533)          OCONV= .TRUE.
(0534)          IF ( OPTO.NE.0. ) OMEGA= OPTO+ .05*(2.-OPTO)
(0535)          GO TO 400
(0536)    350   POPTO= OPTO
(0537)          OPTO= 2./ (1.+ SQRT( 1.-(LEMAX+OMEGA-1.)**2 / (LEMAX*OMEGA**2) )
(0538)          OMEGA= OPTO- (2.-OPTO)/4.
(0539)          IF ( POPTO.EQ.0. ) GO TO 400
(0540)          OCONV= ( ABS(OPTO-POPTO)/ (2.-OPTO) .LT. .05 )
(0541)          IF (OCONV) OMEGA= OPTO
(0542)    C
(0543)    C
(0544)    C TEST CONVERGENCE
(0545)    C
(0546)    400   IF ( TEST.LT.ESOR1 ) GO TO 500
(0547)          IF ( PIT.LT.MAXSOR ) GO TO 200
(0548)    C
(0549)          CALL MSG( 36, 36HPRESSURE ITERATION DID NOT CONVERGE  )
(0550)    C
(0551)    500   CONTINUE
(0552)          RETURN
(0553)    C
(0554)    C-----------------------------------------------------------------------
(0555)    C PROGRAMMING NOTES
(0556)    C
(0557)    C    * EXPANDED MESH  -  AN EXTRA ROW OR COLUMN IS ADDED TO EACH EDGE
(0558)    C      OF THE SQUARE ARRAY IN ORDER TO FACILITATE IMPLEMENTATION OF
(0559)    C      THE NEUMANN BOUNDARY CONDITIONS.  THE PRESSURE VALUES IN THE
(0560)    C      FICTITIOUS CELLS OUTSIDE THE TRUE MESH ARE UPDATED AFTER EACH
(0561)    C      SOR ITERATION.  FOR AN ARRAY STORED ON THE EXPANDED MESH THE
(0562)    C      ELEMENT WITH INDICES (I+1,J+1) CORRESPONDS TO THE FUNCTION
(0563)    C      EVALUATED AT THE TRUE MESH POINT (X(I),Y(J)).
(0564)    C    * DIRICHLET CONDITIONS  -  THE DIRICHLET BOUNDARY CONDITION
(0565)    C      IS IMPLEMENTED BY STORING THE FUNCTION VALUE IN THE P ARRAY IN
(0566)    C      SUBROUTINE INIT.  THE VALUES OF P ALONG THE EDGES WITH DIRICHLET
(0567)    C      CONDITIONS ARE NOT CHANGED IN PSOLVE OR ELSEWHERE.
(0568)    C    * CONVERGENCE - THIS VERSION TESTS THE MAXIMUM RELATIVE CHANGE
(0569)    C      IN PRESSURE AT INTERIOR POINTS.
(0570)    C
(0571)          END
PROGRAM SIZE:   PROCEDURE - 001017    LINKAGE - 000153    STACK - 000034
0000 ERRORS [<PSOLVE>FTN-REV15.3]
```

```
      SUBROUTINE SSICON ( NTRY, AGAIN )

(0572)       SUBROUTINE SSICON ( NTRY, AGAIN )
(0573) C
(0574) C CONTROLS ITERATION TO STEADY STATE SOLUTION.
(0575) C
(0576) C-----------------------------------------------------------------
(0577) C ARGUMENT LIST
(0578) C     NTRY     (INPUT)      1  INITIALIZE ITERATION SCHEME
(0579) C                           2  TEST CONVERGENCE AFTER EACH ITERATION.
(0580) C     AGAIN    (OUTPUT)     WHEN NTRY=2 AGAIN HAS THE FOLLOWING MEANING
(0581) C                           AGAIN=TRUE   SHOULD CONTINUE ITERATIONS
(0582) C                           AGAIN=FALSE  CONVERGENCE OR MAXSSI EXCEEDED
(0583)                             LOGICAL AGAIN
(0584) C-----------------------------------------------------------------
(0584) C COMMON BLOCKS                              (SEE INIT FOR DOCUMENTATION
(0584) C-----------------------------------------------------------------
(0585) C LOCAL DECLARATIONS
(0586)       DIMENSION OLDGL(1)
(0587)       EQUIVALENCE (SCR1,OLDGL)
(0588) C-----------------------------------------------------------------
(0589) C
(0590) C NTRY=1
(0591) C INITIALIZATION ENTRY
(0592) C
(0593)       IF (NTRY.GT.1) GO TO 500
(0594) C
(0595)       ITSSI= 1
(0596)       DO 200 J=1,NJ
(0597)       DO 200 I=1,NI
(0598)       OLDGL(I+(J-1)*NI)= GL(I,J)
(0599)  200  CONTINUE
(0600)       ESORI=EPSSOR*10.
(0601)       RETURN
(0602) C
(0603) C NTRY=2
(0604) C TEST CONVERGENCE
(0605) C
(0606)  500  AGAIN= .FALSE.
(0607)       TEST= 0.
(0608)       DO 550 J=1,NJ
(0609)       DO 550 I=1,NI
(0610)       IJ= I+(J-1)*NI
(0611)       TEST= AMAX1( TEST,
(0612)      *             2.*ABS( (GL(I,J)-OLDGL(IJ))/(GL(I,J)+OLDGL(IJ)) ) )
(0613)  550  CONTINUE
(0614)       WRITE (CRT,560) ITSSI, PIT, TEST
(0615)  560  FORMAT( 1X, 9HITERATION, I3, 3H, (, I3, 18H PRESSURE CYCLES),,
(0616)      *        19H CONVERGENCE TEST =,1PE10.3 )
(0617)       IF (TEST.LT.EPSSSI) RETURN
(0618) C
(0619) C CHECK MAXIMUM NUMBER OF ITERATIONS
(0620) C
(0621)       IF (ITSSI.LT.MAXSSI) GO TO 650
(0622)       CALL MSG(40, 40HSTEADY STATE SOLUTION DID NOT CONVERGE   )
(0623)       RETURN
(0624) C
(0625) C INITIALIZE NEXT ITERATION
(0626) C
(0627)  650  AGAIN= .TRUE.
(0628)       ITSSI= ITSSI+1
(0629)       DO 700 J=1,NJ
```

```
      SUBROUTINE SSICON ( NTRY, AGAIN )

(0630)            DO 700 I=1,NI
(0631)            OLDGL(I+(J-1)*NI)= GL(I,J)
(0632)     700    CONTINUE
(0633)            IF ( TEST .LT. 100.*EPSSSI ) ESCRI= AMIN1( ESORI, 5.*EPSSOR )
(0634)            IF ( TEST .LT.  10.*EPSSSI ) ESORI= EPSSOR
(0635)            RETURN
(0636)      C
(0637)            END
PROGRAM SIZE:   PROCEDURE - 000572    LINKAGE - 000077    STACK - 000026
0000 ERRORS [<SSICON>FTN-REV15.3]
```

SUBROUTINE VLCTY

```
(0638)        SUBROUTINE VLCTY
(0639)  C
(0640)  C CALCULATE VELOCITY FROM DARCYS LAW
(0641)  C
(0642)  C-------------------------------------------------------------------
(0642)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATION)
(0642)  C-------------------------------------------------------------------
(0643)  C STATEMENT FUNCTION
(0644)  C
(0645)        SF(IP,JP)= -KK(IP,JP)/ (VISC*GL(IP,JF))
(0646)  C
(0647)  C-------------------------------------------------------------------
(0648)  C
(0649)  C VX
(0650)  C
(0651)        DO 220 J=1,NJ
(0652)        V(1,1, J)= SF(1, J)* DPDXE(J)
(0653)  C
(0654)        DO 200 I=2,NIM
(0655)        V(1,I, J)= SF(I, J)* ( P(I+2,J+1)-P(I,J+1) )/ (2.*DX)
(0656)   200  CONTINUE
(0657)  C
(0658)        V(1,NI,J)= SF(NI,J)* (P(NI+1,J+1)-P(NI,J+1))/DX
(0659)   220  CONTINUE
(0660)  C
(0661)  C VY
(0662)  C
(0663)        DO 320 I=1,NI
(0664)        V(2,I,1 )= SF(I,1 )* ( DPDY0(I)+(RHL(I,1)-RHL(NI,1))*GRAV )
(0665)  C
(0666)        DO 300 J=2,NJM
(0667)        V(2,I,J )= SF(I,J )*
(0668)       *            ( (P(I+1,J+2)-P(I+1,J))/(2.*DY)+
(0669)       *              (RHL(I,J)-RHL(NI,J))*GRAV )
(0670)   300  CONTINUE
(0671)  C
(0672)        V(2,I,NJ)= SF(I,NJ)* ( DPDYL(I)+(RHL(I,NJ)-RHL(NI,NJ))*GRAV )
(0673)   320  CONTINUE
(0674)  C
(0675)        RETURN
(0676)  C
(0677)  C-------------------------------------------------------------------
(0678)  C PROGRAMMING NOTES
(0679)  C
(0680)  C   * NOTE EXTENDED MESH STORAGE OF P  -  SEE NOTES IN PSOLVE.
(0681)  C-------------------------------------------------------------------
(0682)        END
PROGRAM SIZE:   PROCEDURE - 000604    LINKAGE - 000104    STACK - 000036
0000 ERRORS [<VLCTY >FTN-REV15.3]
```

```
      SUBROUTINE ADB ( SOLVNT, SOLUTE, WPCT, ERRADB )

(0001)        SUBROUTINE ADB ( SOLVNT, SOLUTE, WPCT, ERRADB )
(0002)  C
(0003)  C ACCESS ALLOY DATA BASE.
(0004)  C RETRIEVE LOCAL PHASE DIAGRAM, DENSITIES, VISCOSITY.
(0005)  C ASSUMES DATA BASE WAS SET UP WITH NAME OF SOLVENT FIRST.
(0006)  C
(0007)  C-----------------------------------------------------------------
(0008)  C     SOLVNT   (INPUT)     ALPHANUMERIC NAME OF SOLVENT.
(0009)                             INTEGER SOLVNT
(0010)  C     SOLUTE   (INPUT)     ALPHANUMERIC NAME OF SOLUTE.
(0011)                             INTEGER SOLUTE
(0012)  C     WPCT     (INPUT)     WEIGHT PERCENT OF SOLUTE.
(0013)  C     ERRADB   (OUTPUT)    TRUE WHEN ERROR HAS OCCURRED IN
(0014)  C                          ACCESSING DATA BASE OR WHEN VALUE OF
(0015)  C                          WPCT IS INVALID.
(0016)                             LOGICAL ERRADB
(0017)  C-----------------------------------------------------------------
(0017)  C COMMON BLOCKS                                (SEE INIT FOR DOCUMENTATIOR
(0017)  C-----------------------------------------------------------------
(0018)  C LOCAL DECLARATIONS
(0019)        DIMENSION NAMTRY(2)
(0020)        INTEGER OUTU
(0021)  C-----------------------------------------------------------------
(0022)  C
(0023)        ERRADB= .FALSE.
(0024)        OUTU= CRT
(0025)        IF (BATCH) OUTU= PRNT
(0026)        IF (BATCH) WRITE (PRNT,305)
(0027)        IF (.NOT.BATCH) CALL NEWPAG
(0028)  C
(0029)  C SEARCH FOR SOLVENT-SOLUTE COMBINATION IN DATA BASE.
(0030)  C
(0031)  C     REWIND ADBU      ***** BUG IN PRIME REWIND *****
(0032)        CALL SEARCH ( INTS(4), 6HM1.C.B, INTS(ADBU-4) )
(0033)        CALL SEARCH ( INTS(1), 6HM1.C.B, INTS(ADBU-4) )
(0034)        READ (ADBU,220,END=500,ERR=540) ADBREF
(0035)   200  READ (ADBU,225,END=400,ERR=540) NAMTRY, CLLOW, CLHIGH
(0036)        READ (ADBU,220,END=540,ERR=540) ALREF
(0037)        READ (ADBU,230,END=540,ERR=540) DCLDT, EPR, CE, TE
(0038)        READ (ADBU,230,END=540,ERR=540) DRHDC, RHS, RHLE, RHSE
(0039)        READ (ADBU,230,END=540,ERR=540) VISC
(0040)        IF ( NAMTRY(1).NE.SOLVNT .OR. NAMTRY(2).NE.SOLUTE ) GO TO 200
(0041)        IF ( (WPCT.LT.CLLOW) .OR. (WPCT.GT.CLHIGH) ) GO TO 200
(0042)  C
(0043)   220  FORMAT( 20A4 )
(0044)   225  FORMAT( A4,6X, A4,6X, 2E10.4 )
(0045)   230  FORMAT( 8E10.4 )
(0046)  C
(0047)  C ALLOY LOCATED, DISPLAY DATA TO BE USED IN CALCULATION.
(0048)  C
(0049)        WRITE (OUTU,315) ADBREF
(0050)        WRITE (OUTU,320) SOLVNT, SOLUTE, CLLOW, CLHIGH, SOLUTE, ALREF
(0051)        WRITE (OUTU,340) DCLDT, EPR, CE, TE
(0052)        WRITE (OUTU,345) DRHDC, RHS, RHLE, RHSE
(0053)        WRITE (OUTU,350) VISC
(0054)  C
(0055)        IF (BATCH) RETURN
(0056)  C
(0057)   355  WRITE (CRT,360)
(0058)   360  FORMAT ( /// 18HENTER P TO PROCEED)
```

```
      SUBROUTINE ADB ( SOLVNT, SOLUTE, WPCT, ERRADB )

(0059)        READ (CRT,FMTA4) IRSP
(0060)        IF (IRSP.EQ.4HP   ) RETURN
(0061)        WRITE (CRT,FMT5)
(0062)        GO TO 355
(0063)  C
(0064)   305  FORMAT (1H1)
(0065)   315  FORMAT (//1X, 20HALLOY DATA BASE  -  , 20A4 )
(0066)   320  FORMAT (//1X, 25HSOURCE OF INFORMATION FOR, 2X, A4, 1H-, A4,
(0067)       *              5X, 1PE10.3, 4H TO , 1PE10.3, 10H WT. PCT. , A4,
(0068)       *        / (4X, 20A4) )
(0069)   340  FORMAT (//1X, 13HPHASE DIAGRAM,
(0070)       *        /1X, 45H   TEMPERATURE-COMPOSITION SLOPE
(0071)       *              3X, 1PE10.3, 19H PCT SOLUTE / DEG C,
(0072)       *        /1X, 45H   EQUILIBRIUM PARTITION RATIO
(0073)       *              3X, 1PE10.3,
(0074)       *        /1X, 45H   EUTECTIC COMPOSITION
(0075)       *              3X, 1PE10.3, 11H PCT SOLUTE,
(0076)       *        /1X, 45H   EUTECTIC TEMPERATURE
(0077)       *              3X, 1PE10.3, 6H DEG C,
(0078)       *        /)
(0079)   345  FORMAT (//1X,  9HDENSITIES,
(0080)       *        /1X, 45H   COMPOSITION-DENSITY SLOPE
(0081)       *              3X, 1PE10.3, 24H (GM/CM**3) / PCT SOLUTE,
(0082)       *        /1X, 45H   SOLID DENSITY
(0083)       *              3X, 1PE10.3, 9H GM/CM**3,
(0084)       *        /1X, 45H   LIQUID EUTECTIC DENSITY
(0085)       *              3X, 1PE10.3, 9H GM/CM**3,
(0086)       *        /1X, 45H   SOLID EUTECTIC DENSITY
(0087)       *              3X, 1PE10.3, 9H GM/CM**3,
(0088)       *        /)
(0089)   350  FORMAT (//1X, 45HVISCOSITY
(0090)       *              3X, 1PE10.3, 12H GM/(CM*SEC),
(0091)       *        /)
(0092)  C
(0093)  C SOLVENT-SOLUTE COMBINATION NOT IN DATA BASE FOR COMPOSITION RANGE
(0094)  C BRACKETING WPCT.
(0095)  C PRINT ERROR MESSAGE AND ABORT BATCH JOB OR RETURN CONTROL TO
(0096)  C INTERACTIVE OPERATOR.
(0097)  C
(0098)   400  ERRADB= .TRUE.
(0099)        WRITE (OUTU,420) SOLVNT, WPCT, SOLUTE, ADBREF
(0100)   420  FORMAT (  1X,  22H*****UNABLE TO LOCATE , A4, 1PE10.3, 2H -, A4,
(0101)       *              14H IN DATA BASE.,
(0102)       *        //1X, 29H     DATA BASE IDENTIFIER IS:, 5X, 20A4,
(0103)       *        //1X, 42H     ALLOYS IN DATA BASE ARE LISTED BELOW:,
(0104)       *        //)
(0105)  C     REWIND ADBU     ***** BUG IN PRIME REWIND *****
(0106)        CALL SEARCH ( INTS(4), 6HM1.D.B, INTS(ADBU-4) )
(0107)        CALL SEARCH ( INTS(1), 6HM1.D.B, INTS(ADBU-4) )
(0108)        READ (ADBU,220,END=500,ERR=540)
(0109)   430  READ (ADBU,225,END=460,ERR=540) NAMTRY, CLLOW, CLHIGH
(0110)        READ (ADBU,220,END=460,ERR=540) ALREF
(0111)        DO 435 IR=1,3
(0112)   435  READ (ADBU,220,END=460,ERR=540)
(0113)        WRITE (OUTU,440) NAMTRY, CLLOW, CLHIGH, NAMTRY(2)
(0114)   440  FORMAT ( 6X, A4, 1H-, A4, 5X, 1PE10.3, 4H TO , 1PE10.3,
(0115)       *        10H WT. PCT. , A4 )
(0116)        GO TO 430
(0117)   460  RETURN
(0118)  C
```

```
      SUBROUTINE ADB ( SOLVNT, SOLUTE, WPCT, ERRADB )


(0119)  C ERROR IN DATA BASE SETUP.
(0120)  C
(0121)   500  WRITE (OUTU,520)
(0122)   520  FORMAT ( //1X, 31H***** ALLOY DATA BASE IS EMPTY. )
(0123)        CALL EXIT
(0124)  C
(0125)   540  WRITE (OUTU,560)
(0126)   560  FORMAT ( //1X, 46H***** MISSING DATA OR FORMAT ERROR IN DATABASE
(0127)        CALL EXIT
(0128)  C
(0129)        END
PROGRAM SIZE:   PROCEDURE - 002421    LINKAGE - 000133    STACK - 000030
0000 ERRORS [<ADB   >FTN-REV15.3]
```

```
      SUBROUTINE ADISP ( DAS, ARRAY, N1, N2, N3, IJE, NC, NAME )

(0130)       SUBROUTINE ADISP ( DAS, ARRAY, N1, N2, N3, IJE, NC, NAME )
(0131) C
(0132) C DISPLAYS ARRAY
(0133) C
(0134) C-----------------------------------------------------------------
(0135) C ARGUMENT LIST
(0136) C
(0137) C     DAS       (INPUT)   TRUE WHEN ARRAY IS TO BE DISPLAYED ON CRT
(0138) C                         SCREEN.  FALSE WHEN ARRAY IS TO BE PRINTED.
(0139)                           LOGICAL DAS
(0140) C     ARRAY     (INPUT)   ARRAY TO DISPLAY.
(0141)                           DIMENSION ARRAY (N1,N2,N3)
(0142) C     N1,N2,N3  (INPUT)   DIMENSIONS OF ARRAY.
(0143) C                         N1 IS 1 FOR SCALAR ARRAYS.
(0144) C                              2 FOR VECTOR ARRAYS SUCH AS V.
(0145) C                         N2 IS THE X DIMENSION.
(0146) C                         N3 IS THE Y DIMENSION.
(0147) C     IJE       (INPUT)   EXTENDED MESH FLAG  -  SEE NOTES IN PSOLVE
(0148) C                         0  USUAL MESH
(0149) C                         1  EXTENDED MESH
(0150) C     NC        (INPUT)   NUMBER OF CHARACTERS IN NAME
(0151) C     NAME      (INPUT)   TITLE TO DISPLAY OVER ARRAY
(0152)                           DIMENSION NAME (NC)
(0153) C-----------------------------------------------------------------
(0153) C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0153) C-----------------------------------------------------------------
(0154) C LOCAL DECLARATIONS
(0155)       INTEGER STAR, OUTU
(0156)       DATA NPL /10/,
(0157)      *     STAR /1H*/
(0158) C-----------------------------------------------------------------
(0159) C
(0160)       OUTU= PRNT
(0161)       IF (DAS) OUTU= CRT
(0162)       IF (.NOT.DAS) WRITE (PRNT,200)
(0163)       IF (DAS) CALL NEWPAG
(0164)       WRITE (OUTU,FMT80) TITLE
(0165)       NN= NC/NCW
(0166)       IF ( NN*NCW .LT. NC ) NN= NN+1
(0167)       WRITE (OUTU,220) ( NAME(IN), IN=1,NN )
(0168)       NLNS= 4 + (NN/20)* 3
(0169)       IF ( NN-(NN/20) .GT. 0 ) NLNS= NLNS+ 3
(0170) C
(0171) C NDX ALLOWS FOR 1-D ARRAY (Y VARIATION ONLY).
(0172) C NBLK IS THE NUMBER OF PRINTED BLOCKS THE ARRAY WILL OCCUPY.
(0173) C NPL IS THE MAXIMUM NUMBER OF X VALUES PER LINE.
(0174) C
(0175)       NDX= NI
(0176)       IF ( N2.EQ.1 ) NDX= 1
(0177)       NBLK= NDX/NPL
(0178)       IF ( NPL*NBLK .LT. NDX ) NBLK= NBLK+1
(0179) C
(0180) C LOOP THROUGH PRINTED BLOCKS.
(0181) C
(0182)       DO 500 IBLK=1,NBLK
(0183)       IF ( (.NOT.DAS) .OR. (NLSCRN-NLNS .GT. 0) ) GO TO 300
(0184)       CALL WAIT
(0185)       CALL NEWPAG
(0186)       NLNS= 0
(0187)   300 NLNS= NLNS+ 3
```

```
      SUBROUTINE ADISP ( DAS, ARRAY, N1, N2, N3, IJE, NC, NAME )

(0188)        IF (NDX.NE.1) NLNS= NLNS+ 2
(0189)        IL= (IBLK-1)*NPL +1
(0190)        IU= MIN0( IL+NPL-1, NDX )
(0191)        IF (NDX.NE.1) WRITE (OUTU,320) (XX(I),I=IL,IU)
(0192)        IF (NDX.EQ.1) WRITE (OUTU,325)
(0193)        WRITE (OUTU,340) (STAR, I=IL,IU)
(0194)  C
(0195)  C WITHIN EACH BLOCK, LOOP THROUGH Y FROM TOP OF INGOT TO BOTTOM.
(0196)  C
(0197)        DO 480 JJ=1,NJ
(0198)        IF ( (.NOT.DAS) .OR. (NLSCRN-NLNS .GE. N1) ) GO TO 350
(0199)        CALL WAIT
(0200)        CALL NEWPAG
(0201)        NLNS= 0
(0202)   350  NLNS= NLNS+ 1
(0203)        J= NJ-JJ+1
(0204)        WRITE (OUTU,420) YY(J), (ARRAY(1,  I+IJE,J+IJE), I=IL,IU)
(0205)  C
(0206)        IF (N1.EQ.1) GO TO 480
(0207)        NLNS= NLNS+ N1
(0208)        DO 470 IN1=2,N1
(0209)        WRITE (OUTU,440)        (ARRAY(IN1,I+IJE,J+IJE), I=IL,IU)
(0210)   470  CONTINUE
(0211)        WRITE (OUTU,445)
(0212)  C
(0213)   480  CONTINUE
(0214)  C
(0215)   500  CONTINUE
(0216)  C
(0217)        IF ( DAS .AND. (NLNS.GT.0) ) CALL WAIT
(0218)  C
(0219)        RETURN
(0220)  C
(0221)  C-----------------------------------------------------------------------
(0222)   200  FORMAT (1H1)
(0223)   220  FORMAT ( // (1X,20A4) )
(0224)   320  FORMAT ( //1X, 7H Y/L   *, 50X, 14H(X-XE)/(XL-XC),
(0225)      *           /1X, 7H       *, 10(1X,0PF5.2,5X) )
(0226)   325  FORMAT (  /1X, 7H Y/L   * )
(0227)   340  FORMAT (   1X, 7H  *    *, 10(3X,A1,7X) )
(0228)   420  FORMAT (   1X, 0PF5.2,2H *, 10(1X,1PE10.3) )
(0229)   440  FORMAT (   1X,    7H       *, 10(1X,1PE10.3) )
(0230)   445  FORMAT (/)
(0231)        END
PROGRAM SIZE:   PROCEDURE - 001206    LINKAGE - 000124    STACK - 000046
0000 ERRORS [<ADISP >FTN-REV15.3]
```

```
      SUBROUTINE BATCHI (FIRST)

(0232)        SUBROUTINE BATCHI (FIRST)
(0233)  C
(0234)  C READS BATCH INPUT CARDS.
(0235)  C PRINTS INPUT VALUES.
(0236)  C CHECKS FOR INPUT ERRORS (FULL SCAN BEFORE ABORT).
(0237)  C TERMINATES JOB NORMALLY IF NO NEW INPUT CASE.
(0238)  C
(0239)  C-----------------------------------------------------------------
(0240)  C     FIRST    (INPUT)   TRUE FOR FIRST CASE ONLY.
(0241)                           LOGICAL FIRST
(0242)  C-----------------------------------------------------------------
(0242)  C COMMON BLOCKS                                (SEE INIT FOR DOCUMENTATIOI
(0242)  C-----------------------------------------------------------------
(0243)  C LOCAL DECLARATIONS
(0244)        LOGICAL ERROR, ERRADB
(0245)        INTEGER EFMT (15)
(0246)  C-----------------------------------------------------------------
(0247)  C
(0248)        ERROR= .FALSE.
(0249)        IF (FIRST) CALL SETHOL( EFMT, 60, 60H(1X,50H***** FORMAT ERROR DI
(0250)       *ECTED WHILE READING NEXT ITEM ) )
(0251)  C
(0252)  C READ CASE NAME
(0253)  C
(0254)   200  READ (CARD,210,END=1200) (TITLE(IT,2),IT=1,20)
(0255)        ERROR= .FALSE.
(0256)   210  FORMAT (20A4)
(0257)  C
(0258)  C PRINT PROGRAM IDENTIFICATION PAGE
(0259)  C
(0260)        WRITE (PRNT,220) MODEL
(0261)   220  FORMAT (1H1,  29HMATERIALS PROCESSING IN SPACE,
(0262)      *         //1X, 35HMACROSEGREGATION IN A CASTING INGOT,
(0263)      *         //1X,  5HMODEL, I3,
(0264)      *          /1X, 44H * UNIDIRECTIONAL SOLIDIFICATION OF A BINARY,
(0265)      *                6H ALLOY,
(0266)      *          /1X, 24H * STEADY STATE SOLUTION,
(0267)      *          /1X, 43H * PLANAR ISOTHERMS, RECTANGULAR MUSHY ZONE,
(0268)      *          /1X, 26H * TEMPERATURE FIELD INPUT,
(0269)      *          /1X, 31H * NO CONVECTION IN BULK LIQUID,
(0270)      *          /1X, 40H * ISOTROPIC PERMEABILITY K= GAMMA*GL**2,
(0271)      *          /)
(0272)  C
(0273)  C READ ALLOY
(0274)  C
(0275)        READ (CARD,FMTA4,END=1100) ALNAM(1)
(0276)        READ (CARD,FMTA4,ERR=250,END=1100) ALNAM(2)
(0277)        READ (CARD,*,ERR=250,END=1100) CL0
(0278)        GO TO 300
(0279)   250  WRITE (PRNT,EFMT)
(0280)        ERROR= .TRUE.
(0281)  C
(0282)  C SET UP CASE TITLE BLOCK.
(0283)  C
(0284)   300  ENCODE ( 80, TIFMT, TITLE ) ALNAM(1), CL0, ALNAM(2), MODEL
(0285)        CALL TIMESA( TITLE(14,1) )
(0286)        CALL DATESA( TITLE(17,1) )
(0287)        WRITE (PRNT,FMTA0) TITLE
(0288)  C
(0289)  C GET ALLOY INFORMATION FROM DATA BASE.
```

```
      SUBROUTINE BATCHI (FIRST)
(0290)  C
(0291)        CALL ADH ( ALNAM(1), ALNAM(2), CLO, ERRADB )
(0292)        ERROR= ERROR.OR.ERRADB
(0293)  C
(0294)  C READ AND PRINT REMAINING INPUT PARAMETERS.
(0295)  C LOOPS THROUGH DATA IN ORDER DEFINED BY KEYS IN INCON.
(0296)  C SCANS ALL INPUT, CHECKING FOR FORMAT ERRORS AND FOR PARAMETER
(0297)  C BOUND VIOLATIONS.
(0298)  C
(0299)        WRITE (PRNT,505)
(0300)        WRITE (PRNT,FMT80) TITLE
(0301)        WRITE (PRNT,510)
(0302)   505  FORMAT (1H1)
(0303)   510  FORMAT ( /1X, 10HCASE INPUT, // )
(0304)  C
(0305)        IPGPH= 0
(0306)   520     IPGPH= IPGPH+1
(0307)           IF ( KEYS(1,IPGPH) .EQ. 0 ) GO TO 1000
(0308)           WRITE (PRNT,FMT80) (PGHDR(IP,IPGPH),IP=1,20)
(0309)           ITEM= 0
(0310)   530        ITEM= ITEM+1
(0311)              KEY= KEYS( ITEM, IPGPH )
(0312)              IF ( KEY .EQ. 0 ) GO TO 520
(0313)              IF ( TYPE(KEY) .EQ. 2 ) GO TO 555
(0314)  C
(0315)  C        *** INPUT INTEGER VALUE ***
(0316)              READ (CARD,*,ERR=540,END=1100) IVLU(1,KEY)
(0317)              IF ( (IVLU(1,KEY).GE.IVLL(2,KEY)) .AND.
(0318)         *         (IVLU(1,KEY).LE.IVLL(3,KEY)) ) GO TO 550
(0319)              WRITE (PRNT,IFMT4) IVLL(2,KEY), IVLU(3,KEY)
(0320)              ERROR= .TRUE.
(0321)              GO TO 550
(0322)   540       WRITE (PRNT,EFMT)
(0323)              ERROR= .TRUE.
(0324)   550       WRITE (PRNT,IFMT1) ITEM, (LN(ILN,KEY),ILN=1,NLN),
(0325)         *                        IVLL(1,KEY), (LU(ILU,KEY),ILU=1,NLU)
(0326)              GO TO 530
(0327)  C
(0328)  C        *** INPUT REAL VALUE ***
(0329)   555       READ (CARD,*,ERR=560,END=1100) RVLU(1,KEY)
(0330)              IF ( (RVLU(1,KEY).GE.RVLL(2,KEY)) .AND.
(0331)         *         (RVLU(1,KEY).LE.RVLU(3,KEY)) ) GO TO 570
(0332)              WRITE (PRNT,RFMT4) RVLL(2,KEY), RVLU(3,KEY)
(0333)              ERROR= .TRUE.
(0334)              GO TO 570
(0335)   560       WRITE (PRNT,EFMT)
(0336)              ERROR= .TRUE.
(0337)   570       WRITE (PRNT,RFMT1) ITEM, (LN(ILN,KEY),ILN=1,NLN),
(0338)         *                        RVLU(1,KEY), (LU(ILU,KEY),ILU=1,NLU)
(0339)              GO TO 530
(0340)  C
(0341)  C TERMINATE JOB OR RETURN TO INCON.
(0342)  C
(0343)   1000 IF (.NOT.ERROR) RETURN
(0344)  C
(0345)        WRITE (PRNT,1020)
(0346)   1020 FORMAT ( //1X, 43H***** SCAN OF INPUT FOR THIS CASE COMPLETE.,
(0347)         *         /1X,  45H      CASE ABORTED DUE TO ERRORS NOTED ABOVE. )
(0348)        GO TO 200
(0349)  C
```

```
      SUBROUTINE BATCHI (FIRST)

(0350)    1100 WRITE (PRNT,1120)
(0351)    1120 FORMAT ( //1X, 45H***** RUN ABORTED DUE TO MISSING INPUT CARDS. )
(0352)         CALL EXIT
(0353)   C
(0354)    1200 IF ( FIRST .AND. .NOT.ERROR ) GO TO 1100
(0355)         IF (.NOT.ERROR) WRITE (PRNT,1220)
(0356)    1220 FORMAT ( //1X, 24H***** NORMAL TERMINATION )
(0357)         CALL EXIT
(0358)   C
(0359)         END
PROGRAM SIZE:   PROCEDURE - 002144    LINKAGE - 000225   STACK - 000020
0000 ERRORS [<BATCHI>FTN-REV15.3]
```

```
      SUBROUTINE IACTI (FIRST)

(0360)        SUBROUTINE IACTI (FIRST)
(0361)  C
(0362)  C CONTROLS INTERACTIVE INPUT.
(0363)  C
(0364)  C-----------------------------------------------------------------
(0365)  C     FIRST    (INPUT)   TRUE FOR FIRST CASE ONLY.
(0366)                           LOGICAL FIRST
(0367)  C-----------------------------------------------------------------
(0367)  C COMMON BLOCKS                          (SEE INIT FOR DOCUMENTATION
(0367)  C-----------------------------------------------------------------
(0368)  C LOCAL DECLARATIONS
(0369)        LOGICAL ERRADR
(0370)        DIMENSION IRSP(1)
(0371)  C-----------------------------------------------------------------
(0372)  C
(0373)  C INITIALIZE TERMINAL CONTROL SYSTEM.
(0374)  C
(0375)        IF (.NOT.FIRST) GO TO 200
(0376)        CALL INITT(1200)
(0377)        CALL TERM(3,1024)
(0378)        CALL CHRSIZ(4)
(0379)  C
(0380)  C PUT PROGRAM ID PAGE ON SCREEN.
(0381)  C
(0382)        WRITE (CRT,180) MODEL
(0383)   180  FORMAT (  1X, 29HMATERIALS PROCESSING IN SPACE,
(0384)       *         //1X, 35HMACROSEGREGATION IN A CASTING INGOT,
(0385)       *         //1X,  5HMODEL, I3,
(0386)       *          /1X, 44H * UNIDIRECTIONAL SOLIDIFICATION OF A BINARY,
(0387)       *                  6H ALLOY,
(0388)       *          /1X, 24H * STEADY STATE SOLUTION,
(0389)       *          /1X, 43H * PLANAR ISOTHERMS, RECTANGULAR MUSHY ZONE,
(0390)       *          /1X, 26H * TEMPERATURE FIELD INPUT,
(0391)       *          /1X, 31H * NO CONVECTION IN BULK LIQUID,
(0392)       *          /1X, 40H * ISOTROPIC PERMEABILITY K= GAMMA*GL**2,
(0393)       *          /)
(0394)  C
(0395)  C GET CASE TITLE
(0396)  C
(0397)   200  IF (.NOT.FIRST) CALL NEWPAG
(0398)        WRITE (CRT,210)
(0399)   210  FORMAT ( //// 38HENTER CASE TITLE (UP TO 80 CHARACTERS),
(0400)       *           / 80(1H,) )
(0401)        READ (CRT,220) (TITLE(IT,2),IT=1,20)
(0402)   220  FORMAT (20A4)
(0403)  C
(0404)  C ALLOY INPUT
(0405)  C
(0406)   300  CALL NEWPAG
(0407)   305  WRITE (CRT,310) ALNAM(1), ALNAM(2), CL0
(0408)   310  FORMAT (// 5HALLOY,
(0409)       *         // 5X, 8HSOLVENT:, 2X, A4,
(0410)       *         /  5X, 8HSOLUTE: , 2X, A4, 17H  WEIGHT PERCENT:, 1PE10.
(0411)        WRITE (CRT,320)
(0412)   320  FORMAT (/  26HENTER A TO CHANGE ALLOY OR,
(0413)       *           19H       P TO PROCEED. )
(0414)        IRSP(1)= 4H
(0415)   321  READ (CRT,FMTA4,ERR=322) IRSP
(0416)        IF ( IRSP(1) .EQ. 4HP    ) GO TO 400
(0417)        IF ( IRSP(1) .EQ. 4HA    ) GO TO 325
```

```
      SUBROUTINE IACTI (FIRST)

(0418)    322  WRITE (CRT,FMT5)
(0419)         GO TO 321
(0420)    325  WRITE (CRT,330)
(0421)    330  FORMAT (   38H....ENTER SOLVENT (UP TO 4 CHARACTERS) )
(0422)         READ (CRT,FMTA4) ALNAM(1)
(0423)         WRITE (CRT,340)
(0424)    340  FORMAT (   37H....ENTER SOLUTE (UP TO 4 CHARACTERS) )
(0425)         READ (CRT,FMTA4) ALNAM(2)
(0426)    345  WRITE (CRT,350)
(0427)    350  FORMAT (   20HENTER WEIGHT PERCENT )
(0428)         READ (CRT,*,ERR=360) CLO
(0429)         GO TO 400
(0430)    360  WRITE (CRT,FMT5)
(0431)         GO TO 345
(0432)  C
(0433)  C GET ALLOY INFORMATION FROM DATA BASE.
(0434)  C
(0435)    400  CALL ADB ( ALNAM(1), ALNAM(2), CLO, ERRADB )
(0436)         IF (ERRADB) GO TO 305
(0437)  C
(0438)  C SET UP CASE TITLE
(0439)  C
(0440)         ENCODE ( 80, TIFMT, TITLE ) ALNAM(1), CLO, ALNAM(2), MODEL
(0441)         CALL TIME$A( TITLE(14,1) )
(0442)         CALL DATE$A( TITLE(17,1) )
(0443)  C
(0444)  C LOOP THROUGH INPUT SELECTION PAGES.
(0445)  C EACH PAGE IS A SEPARATE DISPLAY. VARIABLES INCLUDED IN EACH PAGE
(0446)  C ARE DETERMINED BY ARRAY KEYS SET UP IN INCOR.  A ZERO IN KEYS
(0447)  C INDICATES THE END OF LIST FOR THE PAGE.  A ZERO IN KEYS(1,IPAGE)
(0448)  C INDICATES THE END OF PAGES.
(0449)  C LN AND LU ARE VARIABLE-NAME AND VARIABLE-UNITS LABELS.
(0450)  C (I/R)VLU( 1,2,3, KEY) ARE THE VALUE, LOWER BOUND, AND UPPER BOUND,
(0451)  C RESPECTIVELY OF INPUT VARIABLE NUMBER KEY.
(0452)  C
(0453)         IPAGE= 0
(0454)    600     IPAGE= IPAGE+1
(0455)            IF ( KEYS(1,IPAGE) .EQ. 0 ) GO TO 800
(0456)    610     CALL NEWPAG
(0457)            WRITE (CRT,FMT80) TITLE, (PCHDR(IP,IPAGE),IP=1,20)
(0458)            ITEM=0
(0459)    620        ITEM= ITEM+1
(0460)               KEY= KEYS(ITEM,IPAGE)
(0461)               IF (KEY.EQ.0) GO TO 630
(0462)               IF ( TYPE(KEY) .EQ. 1 )
(0463)         *       WRITE (CRT,IFMT1) ITEM, (LN(ILN,KEY),ILN=1,NLN),
(0464)         *                         IVLU(1,KEY), (LU(ILU,KEY),ILU=1,NLU)
(0465)               IF ( TYPE(KEY) .EQ. 2 )
(0466)         *       WRITE (CRT,RFMT1) ITEM, (LN(ILN,KEY),ILN=1,NLN),
(0467)         *                         RVLU(1,KEY), (LU(ILU,KEY),ILU=1,NLU)
(0468)               GO TO 620
(0469)  C
(0470)    630     WRITE (CRT,640)
(0471)    640     FORMAT ( /// 31HENTER ITEM NUMBER TO CHANGE, OR
(0472)         *             / 29H      P  ----------TO PROCEED. )
(0473)            IRSP(1)= 4H
(0474)            READ (CRT,FMTA4,ERR=650) IRSP
(0475)            IF (IRSP(1).EQ.4HP   ) GO TO 800
(0476)            IF (IRSP(1).EQ.4H    ) GO TO 650
(0477)            DECODE ( NCW, *, IRSP, ERR=650 ) ITEMC
```

```
      SUBROUTINE IACTI (FIRST)

(0478)            IF ( (ITEMC.LT.ITEM) .AND. (ITEMC.GT.0) ) GO TO 655
(0479)   650      WRITE (CRT,FMT5)
(0480)            GO TO 630
(0481)  C
(0482)   655      KEY= KEYS(ITEMC,IPAGE)
(0483)            IF ( TYPE(KEY) .EQ. 2 ) GO TO 680
(0484)   660      WRITE (CRT,FMT2) (LN(ILN,KEY),ILN=1,NLN)
(0485)            READ (CRT,*,ERR=670) IVLU(1,KEY)
(0486)            IF ( (IVLU(1,KEY).GE.IVLU(2,KEY)) .AND.
(0487)           *      (IVLU(1,KEY).LE.IVLU(3,KEY)) ) GO TO 610
(0488)            WRITE (CRT,IFMT4) IVLU(2,KEY), IVLU(3,KEY)
(0489)            GO TO 660
(0490)   670      WRITE (CRT,FMT5)
(0491)            GO TO 660
(0492)  C
(0493)   680      WRITE (CRT,FMT2) (LN(ILN,KEY),ILN=1,NLN)
(0494)            READ (CRT,*,ERR=690) RVLU(1,KEY)
(0495)            IF ( (RVLU(1,KEY).GE.RVLU(2,KEY)) .AND.
(0496)           *      (RVLU(1,KEY).LE.RVLU(3,KEY)) ) GO TO 610
(0497)            WRITE (CRT,RFMT4) RVLU(2,KEY),RVLU(3,KEY)
(0498)            GO TO 680
(0499)   690      WRITE (CRT,FMT5)
(0500)            GO TO 680
(0501)  C
(0502)  C PUT MESSAGE ON SCREEN DURING COMPUTATION.
(0503)  C
(0504)   800   CALL NEWPAG
(0505)         WRITE (CRT,810) TITLE
(0506)   810   FORMAT ( 32HCALCULATION IN PROGRESS FOR CASE,
(0507)        *          /// 2(20A4/) // )
(0508)  C
(0509)         RETURN
(0510)         END
PROGRAM SIZE:   PROCEDURE - 002622    LINKAGE - 000204    STACK - 000020
0000 ERRORS [<IACTI >FTN-REV15.3]
```

```
      SUBROUTINE INCON (FIRST)
(0511)        SUBROUTINE INCON (FIRST)
(0512)  C
(0513)  C INPUT CONTROLLER
(0514)  C
(0515)  C-----------------------------------------------------------------
(0516)  C     FIRST    (INPUT)   TRUE FOR FIRST CASE ONLY.
(0517)                           LOGICAL FIRST
(0518)  C-----------------------------------------------------------------
(0518)  C COMMON BLOCKS                          (SEE INIT FOR DOCUMENTATION)
(0518)  C-----------------------------------------------------------------
(0519)  C INITIALIZE KEYS TO INPUT ARRAYS
(0520)  C KEYS(I1,I2) IS THE INDEX OF THE I1 ST ITEM ON PAGE OR PARAGRAPH I2.
(0521)  C ZERO IS AN END-OF-LIST INDICATOR FOR BOTH ROWS AND COLUMNS.
(0522)  C
(0523)        DATA KEYS / 1, 2, 3, 4, 21*0,
(0524)       *            5, 24*0,
(0525)       *            6, 7, 8, 9, 11, 12, 19*0,
(0526)       *            25*0, 25*0 /
(0527)  C-----------------------------------------------------------------
(0528)  C
(0529)  C SET UP STANDARD CASE AND BOUNDS - FIRST CASE ONLY
(0530)  C ALSO SETS UP LABELS FOR BOTH BATCH AND INTERACTIVE DISPLAYS.
(0531)  C
(0532)        IF (.NOT.FIRST) GO TO 200
(0533)        NLN= 10
(0534)        NLU= 5
(0535)        ALNAM(1)= 4HAL
(0536)        ALNAM(2)= 4HCU
(0537)        CLO= 4.5
(0538)        TYPE(1)= 2
(0539)        CALL SETHOL( LN(1,1), 40,
(0540)       *             40HMUSHY ZONE WIDTH                          )
(0541)        CALL SETHOL( LU(1,1), 20, 20H(CM)                          )
(0542)        RVLU(1, 1)= 5.
(0543)        RVLU(2, 1)= 0.
(0544)        RVLU(3, 1)= RINF
(0545)        TYPE(2)= 2
(0546)        CALL SETHOL( LN(1,2), 40,
(0547)       *             40HMUSHY ZONE HEIGHT                         )
(0548)        CALL SETHOL( LU(1,2), 20, 20H(CM)                          )
(0549)        RVLU(1, 2)= 10.
(0550)        RVLU(2, 2)= 0.
(0551)        RVLU(3, 2)= RINF
(0552)        TYPE(3)= 2
(0553)        CALL SETHOL( LN(1,3), 40,
(0554)       *             40HCOOLING RATE                              )
(0555)        CALL SETHOL( LU(1,3), 20, 20H(DEG C/SEC)                   )
(0556)        RVLU(1, 3)= .396
(0557)        RVLU(2, 3)= 0.
(0558)        RVLU(3, 3)= RINF
(0559)        TYPE(4)= 2
(0560)        CALL SETHOL( LN(1,4), 40,
(0561)       *             40HGRAVITATIONAL FORCE                       )
(0562)        CALL SETHOL( LU(1,4), 20, 20H(G)                           )
(0563)        RVLU(1, 4)= 1.
(0564)        RVLU(2, 4)= 0.
(0565)        RVLU(3, 4)= RINF
(0566)        TYPE(5)= 2
(0567)        CALL SETHOL( LN(1,5), 40,
(0568)       *             40HGAMMA                                     )
```

```
      SUBROUTINE INCON (FIRST)
(0569)        CALL SETHOL( LU(1,5), 20, 20H(CM**2)                    )
(0570)        RVLU(1, 5)= 6.E-7
(0571)        RVLU(2, 5)= 0.
(0572)        RVLU(3, 5)= RINF
(0573)        TYPE(6)= 1
(0574)        CALL SETHOL( LN(1,6), 40,
(0575)       *             40HNUMBER OF HORIZONTAL MESH POINTS          )
(0576)        CALL SETHOL( LU(1,6), 20, 20H                           )
(0577)        IVLU(1, 6)= 20
(0578)        IVLU(2, 6)= 10
(0579)        IVLU(3, 6)= IDIM
(0580)        TYPE(7)= 1
(0581)        CALL SETHOL( LN(1,7), 40,
(0582)       *             40HNUMBER OF VERTICAL MESH POINTS            )
(0583)        CALL SETHOL( LU(1,7), 20, 20H                           )
(0584)        IVLU(1, 7)= 20
(0585)        IVLU(2, 7)= 10
(0586)        IVLU(3, 7)= JDIM
(0587)        TYPE(8)= 1
(0588)        CALL SETHOL( LN(1,8), 40,
(0589)       *             40HMAXIMUM NUMBER OF PRESSURE ITERATIONS     )
(0590)        CALL SETHOL( LU(1,8), 20, 20H                           )
(0591)        IVLU(1, 8)= 500
(0592)        IVLU(2, 8)= 1
(0593)        IVLU(3, 8)= 2000
(0594)        TYPE(9)= 2
(0595)        CALL SETHOL( LN(1,9), 40,
(0596)       *             40HPRESSURE CONVERGENCE CRITERION            )
(0597)        CALL SETHOL( LU(1,9), 20, 20H                           )
(0598)        RVLU(1, 9)= 1.E-5
(0599)        RVLU(2, 9)= 0.
(0600)        RVLU(3, 9)= RINF
(0601)        TYPE(11)= 1
(0602)        CALL SETHOL( LN(1,11), 40,
(0603)       *             40HMAX NUMBER OF STEADY-STATE ITERATIONS     )
(0604)        CALL SETHOL( LU(1,11), 20, 20H                          )
(0605)        IVLU(1,11)= 40
(0606)        IVLU(2,11)= 1
(0607)        IVLU(3,11)= 500
(0608)        TYPE(12)= 2
(0609)        CALL SETHOL( LN(1,12), 40,
(0610)       *             40HSTEADY-STATE CONVERGENCE CRITERION        )
(0611)        CALL SETHOL( LU(1,12), 20, 20H                          )
(0612)        RVLU(1,12)= 1.E-4
(0613)        RVLU(2,12)= 5.*ERCSN
(0614)        RVLU(3,12)= RINF
(0615)  C
(0616)  C SET UP PAGE/PARAGRAPH TITLES AND I/O FORMATS
(0617)  C
(0618)        CALL SETHOL( PGHDR(1,1), 80, 80HSOLIDIFICATION PROCESS PARAMETERS
(0619)       *                                                        )
(0620)        CALL SETHOL( PGHDR(1,2), 80, 80HPERMEABILITY MODEL PARAMETERS
(0621)       *                                                        )
(0622)        CALL SETHOL( PGHDR(1,3), 80, 80HNUMERICAL METHODS CONTROL PARAMET
(0623)       *RS                                                      )
(0624)        CALL SETHOL( IFMT1,40,40H( 1X, I2, 5X, 10A4, I10, 5X, 5A4 )
(0625)        CALL SETHOL( RFMT1,40,40H( 1X, I2, 5X, 10A4, 1PE10.3, 5X, 5A4 )
(0626)        CALL SETHOL(  FMT2,20,20H( 7H ENTER , 10A4 )  )
(0627)        CALL SETHOL( IFMT4, 72, 72H( 1X, 42H***** INPUT ERROR, VALUE MUST
(062*       *LIE IN RANGE, I10, 3H TO, I10 )   )
```

```
      SUBROUTINE INCON (FIRST)

(0629)        CALL SETFOL( RFMT4, 80, 80H( 1X, 42H***** INPUT ERROR, VALUE MUST
(0630)       *LIE IN RANGE, 1PE10.3, 3H TO, 1PE10.3 )   )
(0631)        CALL SETFOL(  FMT5, 28, 28H( 22H***** INVALID RESPONSE) )
(0632)        CALL SETHOL( FMT80, 12, 12H( /1X, 20A4) )
(0633)        CALL SETHOL( FMTA4,  8,  8H( A4 )   )
(0634)        CALL SETHOL( TIFMT, 64, 64H( A4, 0PF8.4, 4X, A4, 4X, 20HSOLIDIFIC
(0635)       *TION MODEL, I3, 33X )  -)
(0636)  C
(0637)  C GET BATCH INPUT
(0638)  C
(0639)   200  IF (BATCH) CALL BATCHI(FIRST)
(0640)  C
(0641)  C GET INTERACTIVE INPUT
(0642)  C
(0643)        IF (.NOT.BATCH) CALL IACTI(FIRST)
(0644)  C
(0645)  C EXTRACT VALUES FROM INPUT ARRAY AND CONVERT TO INTERNAL UNITS (CGS).
(0646)  C
(0647)        DXM= RVLU(1,1)
(0648)        DYM= RVLU(1,2)
(0649)        DTDTM= -RVLU(1,3)
(0650)        GFORCE= RVLU(1,4)
(0651)        GRAV= GFORCE*980.66
(0652)        GAMMA= RVLU(1,5)
(0653)        NI= IVLU(1,6)
(0654)        NJ= IVLU(1,7)
(0655)        MAXSOR= IVLU(1,8)
(0656)        EPSSOR= RVLU(1,9)
(0657)        OMEGA= RVLU(1,10)
(0658)        MAXSSI= IVLU(1,11)
(0659)        EPSSSI= RVLU(1,12)
(0660)  C
(0661)        RETURN
(0662)        END
PROGRAM SIZE:   PROCEDURE - 002314    LINKAGE - 000342    STACK - 000016
0000 ERRORS [<INCON >FTN-REV15.3]
```

```
SUBROUTINE MSG ( NCH, MESSGE )

(0663)        SUBROUTINE MSG ( NCH, MESSGE )
(0664)  C
(0665)  C DISPLAYS MESSAGE ON PRINTED OUTPUT OR CRT.
(0666)  C
(0667)  C-----------------------------------------------------------------------
(0668)  C     NCH      (INPUT)    NUMBER OF CHARACTERS IN MESSAGE.
(0669)  C     MESSGE   (INPUT)    MESSAGE TO DISPLAY.
(0670)                            DIMENSION MESSGE(NCH)
(0671)  C-----------------------------------------------------------------------
(0671)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATION
(0671)  C-----------------------------------------------------------------------
(0672)  C LOCAL DECLARATIONS
(0673)        INTEGER OUTU
(0674)  C-----------------------------------------------------------------------
(0675)  C
(0676)        NN= NCH/NCW
(0677)        IF (NN*NCW .LT. NCH) NN=NN+1
(0678)        OUTU= PRNT
(0679)        IF (.NOT.BATCH) OUTU= CRT
(0680)        WRITE (OUTU,100) (MESSGE(IN),IN=1,NN)
(0681)   100  FORMAT ( 1X, 5(1H*), 1X, 20A4 )
(0682)  C
(0683)        RETURN
(0684)        END
PROGRAM SIZE:   PROCEDURE - 000124    LINKAGE - 000045   STACK - 000022
0000 ERRORS [<MSG    >FTN-REV15.3]
```

```
      SUBROUTINE OUTCON (FIRST)
(0685)        SUBROUTINE OUTCON (FIRST)
(0686)  C
(0687)  C OUTPUT CONTROLLER
(0688)  C
(0689)  C-----------------------------------------------------------------------
(0690)  C     FIRST    (INPUT)   TRUE FOR FIRST CASE ONLY.
(0691)                           LOGICAL FIRST
(0692)  C-----------------------------------------------------------------------
(0692)  C COMMON BLOCKS                    (SEE INIT FOR DOCUMENTATION
(0692)  C-----------------------------------------------------------------------
(0693)  C LOCAL DECLARATIONS
(0694)        DIMENSION IRSP(1)
(0695)        DIMENSION BLPR(1)
(0696)        EQUIVALENCE (SCR1,BLPR)
(0697)        LOGICAL PASS1
(0698)  C-----------------------------------------------------------------------
(0699)  C
(0700)        IF (.NOT.BATCH) GO TO 300
(0701)  C
(0702)  C BATCH OUTPUT
(0703)  C
(0704)        CALL ADISP ( .FALSE., T,      1, IDIM,  JDIM,   0,
(0705)       *           20, 20HTEMPERATURE (DEG C)  )
(0706)  C
(0707)        CALL ADISP ( .FALSE., CL,     1, IDIM,  JDIM,   0,
(0708)       *           36, 36HLIQUID COMPOSITION (WT PCT SOLUTE)  )
(0709)  C
(0710)        CALL ADISP ( .FALSE., RHL,    1, IDIM,  JDIM,   0,
(0711)       *           28, 28HLIQUID DENSITY (GM/CM**3)    )
(0712)  C
(0713)        CALL ADISP ( .FALSE., GL,     1, IDIM,  JDIM,   0,
(0714)       *           24, 24HVOLUME FRACTION LIQUID    )
(0715)  C
(0716)        DO 220 J=1,NJ
(0717)        DO 220 I=1,NI
(0718)   220  BLPR(I+(J-1)*NI)= P(I+1,J+1) + RHL(NI,J)*GRAV*(Y(NJ)-Y(J))
(0719)        CALL ADISP ( .FALSE., BLPR,   1, NI,    NJ,     0,
(0720)       *           32, 32HPRESSURE (P-P0) (DYNES/CM**2)    )
(0721)  C
(0722)        CALL ADISP ( .FALSE., V,      2, IDIM,  JDIM,   0,
(0723)       *           16, 16HVELOCITY (CM/S)  )
(0724)  C
(0725)        CALL ADISP ( .FALSE., AVCS,   1, 1,     JDIM,   0,
(0726)       *           40, 40HFINAL LOCAL AVERAGE COMPOSITION (WT PCT) )
(0727)  C
(0728)        RETURN
(0729)  C
(0730)  C INTERACTIVE DISPLAYS - TABLES OR PLOTS.
(0731)  C
(0732)   300  CALL BELL
(0733)        CALL TSEND
(0734)        PASS1= .TRUE.
(0735)   320  IF (.NOT.PASS1) CALL NEWPAG
(0736)        PASS1= .FALSE.
(0737)        WRITE (CRT,340)
(0738)   340  FORMAT ( ///  33HENTER  T TO DISPLAY TABULAR DATA,,
(0739)       *           /  27H       G TO DISPLAY GRAPHS,,
(0740)       *           /  29H       Q TO TERMINATE RUN, OR,
(0741)       *           /  33H       N TO PROCEED TO NEXT CASE. )
(0742)        IRSP(1)= 4H
```

```
      SUBROUTINE OUTCON (FIRST)
(0743)   345  READ (CRT,FMTA4,ERR=350) IRSP
(0744)        IF ( IRSP(1).EQ.4HT    ) GO TO 500
(0745)        IF ( IRSP(1).EQ.4HG    ) GO TO 700
(0746)        IF ( IRSP(1).EQ.4HQ    ) CALL EXIT
(0747)        IF ( IRSP(1).EQ.4HN    ) RETURN
(0748)   350  WRITE (CRT,FMT5)
(0749)        GO TO 345
(0750)   C
(0751)   C DISPLAY TABULAR OUTPUT
(0752)   C
(0753)    500  CALL NEWPAG
(0754)         WRITE (CRT,FMT80) TITLE
(0755)         WRITE (CRT,510)
(0756)    510  FORMAT(' // 16H  1  TEMPERATURE,
(0757)        *         /  24H  2  LIQUID COMPOSITION ,
(0758)        *         /  20H  3  LIQUID DENSITY ,
(0759)        *         /  28H  4  VOLUME FRACTION LIQUID ,
(0760)        *         /  20H  5  PRESSURE (P-P0),
(0761)        *         /  34H  6  PRESSURE - BULK HYDROSTATIC P,
(0762)        *         /  16H  7  VELOCITY  ,
(0763)        *         /  36H  8  FINAL LOCAL AVERAGE COMPOSITION,
(0764)        *         /  35H  9  PRESSURE EQUATION COEFFICIENTS,
(0765)        *         /  31H 10  -LOCAL SOLIDIFICATION RATE )
(0766)    530  WRITE (CRT,535)
(0767)    535  FORMAT( /// 41H ENTER ITEM NUMBER OF TABLE TO DISPLAY, OR,
(0768)        *         /   19H      P TO PROCEED. )
(0769)         IRSP(1)= 4H
(0770)         READ (CRT,FMTA4,ERR=540) IRSP
(0771)         IF (IRSP(1).EQ.4HP    ) GO TO 320
(0772)         IF (IRSP(1).EQ.4H     ) GO TO 540
(0773)         DECODE( NCW, *, IRSP, ERR=540 ) ITEM
(0774)         IF ( ITEM.GT.0 .AND. ITEM.LT.11 ) GO TO 550
(0775)    540  WRITE (CRT,FMT5)
(0776)         GO TO 530
(0777)    550  GO TO (560,570,580,590,600,605,610,620,630,640), ITEM
(0778)   C
(0779)    560  CALL ADISP ( .TRUE., T,      1, IDIM,   JDIM,    0,
(0780)        *              20, 20HTEMPERATURE (DEG C)  )
(0781)         GO TO 500
(0782)   C
(0783)    570  CALL ADISP ( .TRUE., CL,     1, IDIM,   JDIM,    0,
(0784)        *              36, 36HLIQUID COMPOSITION (WT PCT SOLUTE)  )
(0785)         GO TO 500
(0786)   C
(0787)    580  CALL ADISP ( .TRUE., RHL,    1, IDIM,   JDIM,    0,
(0788)        *              28, 28HLIQUID DENSITY (GM/CM**3)    )
(0789)         GO TO 500
(0790)   C
(0791)    590  CALL ADISP ( .TRUE., GL,     1, IDIM,   JDIM,    0,
(0792)        *              24, 24HVOLUME FRACTION LIQUID    )
(0793)         GO TO 500
(0794)   C
(0795)    600  DO 602 J=1,NJ
(0796)         DO 602 I=1,NI
(0797)    602  BLPR(I+(J-1)*NI)= P(I+1,J+1) + RHL(NI,J)*GRAV*(Y(NJ)-Y(J))
(0798)         CALL ADISP ( .TRUE., BLPR,   1, NI,     NJ,      0,
(0799)        *              32, 32HPRESSURE (P-P0) (DYNES/CM**2)    )
(0800)         GO TO 500
(0801)   C
(0802)    605  CALL ADISP ( .TRUE., P,      1, IDIM+2, JDIM+2,  1,
```

```
      SUBROUTINE OUTCON (FIRST)

(0803)      *                44, 44HPRESSURE -BULK HYDROSTATIC P (DYNES/CM**2)  )
(0804)       GO TO 500
(0805)  C
(0806)   610  CALL ADISP ( .TRUE., V,       2, IDIM,   JDIM,   0,
(0807)      *                16, 16HVELOCITY (CM/S)    )
(0808)       GO TO 500
(0809)  C
(0810)   620  CALL ADISP ( .TRUE., AVCS,    1, 1,      JDIM,   0,
(0811)      *                40, 40HFINAL LOCAL AVERAGE COMPOSITION (WT PCT)  )
(0812)       GO TO 500
(0813)  C
(0814)   630  CALL ADISP ( .TRUE., A,       2, IDIM+2, JDIM+2, 1,
(0815)      *                4, 4HA    )
(0816)       CALL ADISP ( .TRUE., B,       1, IDIM+2, JDIM+2, 1,
(0817)      *                4, 4HB    )
(0818)       GO TO 500
(0819)  C
(0820)   640  CALL ADISP ( .TRUE., DGLCTM, 1, IDIM,   JDIM,   0,
(0821)      *                28, 28H-LOCAL SOLIDIFICATION RATE   )
(0822)       GO TO 500
(0823)  C
(0824)  C DISPLAY PLOTS
(0825)  C
(0826)   700  CONTINUE
(0827)       CALL GPHCON
(0828)       GO TO 320
(0829)  C
(0830)       END
PROGRAM SIZE:   PROCEDURE - 002441    LINKAGE - 000150    STACK - 000030
0000 ERRORS [<OUTCON>FTN-REV15.3]
```

SUBROUTINE SETHOL (ARRAY, NCH, HOLCON)

```
(0831)        SUBROUTINE SETHOL ( ARRAY, NCH, HOLCON )
(0832)  C
(0833)  C PUTS HOLLERITH CONSTANT INTO ARRAY.
(0834)  C USED TO AVOID THE CRUDE RESTRICTIONS OF THE ANSI
(0835)  C DATA STATEMENT.
(0836)  C
(0837)  C-------------------------------------------------------------------
(0838)  C ARGUMENT LIST
(0839)  C
(0840)  C     ARRAY     (INPUT)   HOLLERITH STRING WILL BE STORED IN ARRAY.
(0841)                            INTEGER ARRAY (NCH)
(0842)  C     NCH       (INPUT)   NUMBER OF CHARACTERS IN HOLLERITH STRING.
(0843)  C     HOLCON    (INPUT)   HOLLERITH STRING TO STORE.
(0844)                            INTEGER HOLCON (NCH)
(0845)  C-------------------------------------------------------------------
(0845)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATION
(0845)  C-------------------------------------------------------------------
(0846)  C
(0847)        NN= NCH/NCW
(0848)        IF ( NN*NCW .LT. NCH ) NN= NN+1
(0849)        DO 100 IN=1,NN
(0850)        ARRAY(IN)= HOLCON(IN)
(0851)   100  CONTINUE
(0852)  C
(0853)        RETURN
(0854)        END
PROGRAM SIZE:   PROCEDURE - 000057   LINKAGE - 000026   STACK - 000024
0000 ERRORS [<SETHOL>FTN-REV15.3]
```

SUBROUTINE WAIT

```
(0855)          SUBROUTINE WAIT
(0856)    C
(0857)    C WAIT FOR OPERATOR RESPONSE.
(0858)    C
(0859)    C-----------------------------------------------------------------
(0859)    C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATION
(0859)    C-----------------------------------------------------------------
(0860)    C
(0861)     140  READ (CRT,FMTA4) IRSP
(0862)          IF (IRSP.EQ.4HP   ) RETURN
(0863)          WRITE (CRT,FMT5)
(0864)          WRITE (CRT,160)
(0865)     160  FORMAT (18HENTER P TO PROCEED)
(0866)          GO TO 140
(0867)          END
PROGRAM SIZE:   PROCEDURE - 000066    LINKAGE - 000041    STACK - 000012
0000 ERRORS [<WAIT  >FTN-REV15.3]
```

```
      SUBROUTINE AXES ( HMIN, HMAX, HOFF, NHTCS, NHMTCS, NHLAB, HLABEL,
```

```
(0001)       SUBROUTINE AXES ( HMIN, HMAX, HOFF, NHTCS, NHMTCS, NHLAB, HLABEL,
(0002)      *                  VMIN, VMAX, VOFF, NVTCS, NVMTCS, NVLAB1, VLABL1
(0003)      *                  NVLAB2, VLABL2, NVLAB3, VLABL3 )
(0004)  C
(0005)  C DRAW AND LABEL AXES.
(0006)  C
(0007)  C-----------------------------------------------------------------
(0008)  C     HMIN       (INPUT)    MINIMUM VALUE ON HORIZONTAL AXIS.
(0009)  C     HMAX       (INPUT)    MAXIMUM VALUE ON HORIZONTAL AXIS.
(0010)  C     HOFF       (INPUT)    VERTICAL OFFSET OF HORIZONTAL AXIS FROM VWL.
(0011)  C                           RASTER UNITS.
(0012)                              INTEGER HOFF
(0013)  C     NHTCS      (INPUT)    NUMBER OF HORIZONTAL TIC SPACES.
(0014)  C     NHMTCS     (INPUT)    NUMBER OF MINOR TIC SPACES BETWEEN
(0015)  C                           EACH PAIR OF MAJOR TICS.
(0016)  C     NHLAB      (INPUT)    NUMBER OF CHARACTERS IN THE LABEL.
(0017)  C     HLABEL     (INPUT)    AXIS LABEL (ALPHANUMERIC).
(0018)                              INTEGER HLABEL (NHLAB)
(0019)  C     VMIN       (INPUT)    MINIMUM VALUE ON VERTICAL AXIS.
(0020)  C     VMAX       (INPUT)    MAXIMUM VALUE ON VERTICAL AXIS.
(0021)  C     VOFF       (INPUT)    VERTICAL OFFSET OF X AXIS FROM VWL.
(0022)  C                           RASTER UNITS.
(0023)                              INTEGER VOFF
(0024)  C     NVTCS      (INPUT)    NUMBER OF VERTICAL TIC SPACES.
(0025)  C     NVMTCS     (INPUT)    NUMBER OF MINOR TIC SPACES BETWEEN
(0026)  C                           EACH PAIR OF MAJOR TICS.
(0027)  C     NVLAB1     (INPUT)    NUMBER OF CHARACTERS IN VLABL1.
(0028)  C     VLABL1     (INPUT)    AXIS LABEL 1 (ALPHANUMERIC).
(0029)                              INTEGER VLABL1 (NVLAB1)
(0030)  C     NVLAB2     (INPUT)    NUMBER OF CHARACTERS IN VLABL2.
(0031)  C     VLABL2     (INPUT)    AXIS LABEL 2 (ALPHANUMERIC).
(0032)                              INTEGER VLABL2 (NVLAB2)
(0033)  C     NVLAB3     (INPUT)    NUMBER OF CHARACTERS IN VLABL3.
(0034)  C     VLABL3     (INPUT)    AXIS LABEL 3 (ALPHANUMERIC).
(0035)                              INTEGER VLABL3 (NVLAB3)
(0036)  C-----------------------------------------------------------------
(0036)  C COMMON BLOCKS                          (SEE INIT FOR DOCUMENTATION
(0036)  C-----------------------------------------------------------------
(0037)  C
(0038)  C SET UP HORIZONTAL AXIS AND PUT ON LABEL.
(0039)  C
(0040)        CALL XNEAT( 0 )
(0041)        CALL XZERO( 1 )
(0042)        CALL XFRM( 2 )
(0043)        CALL XLEN( VCH )
(0044)        CALL XTICS( NHTCS )
(0045)        CALL XMTCS( NHMTCS )
(0046)        CALL DLIMX( HMIN, HMAX )
(0047)        IHL= HWL+ (HWR-HWL-LINLDT(NHLAB))/2
(0048)        IVL= VWL+ HOFF- 3*VCH- 2
(0049)        CALL MOVABS( IHL, IVL )
(0050)        CALL AOUTST( NHLAB, HLABEL )
(0051)  C
(0052)  C SET UP VERTICAL AXIS AND PUT ON LABELS.
(0053)  C
(0054)        CALL YNEAT( 0 )
(0055)        CALL YZERO( 1 )
(0056)        CALL YFRM( 2 )
(0057)        CALL YLEN( VCH )
(0058)        CALL YTICS( NVTCS )
```

```
      SUBROUTINE AXES ( HMIN, HMAX, HOFF, NHTCS, NHMTCS, NHLAB, HLABEL,
(0059)            CALL YMTCS( NVMTCS )
(0060)            CALL DLIMY( VMIN, VMAX )
(0061)            MNVLAB= MAX0( NVLAB1, NVLAB2, NVLAB3 )
(0062)            IHL= MAX0( 0, HWL+VOFF-LINWDT(MNVLAB)-VCH/2-2 )
(0063)            IVL= VWU - VW/(2*NVTCS)
(0064)            IF ( NVLAB2 .GT. 0 ) IVL= IVL + (VCH+1)/2
(0065)            IF ( NVLAB3 .GT. 0 ) IVL= IVL + (VCH+1)/2
(0066)            CALL MOVABS( IHL, IVL )
(0067)            CALL AOUTST( NVLAB1, VLABL1 )
(0068)            IF ( NVLAB2 .EQ. 0 ) GO TO 200
(0069)            IHL= MAX0( 0, HWL+VOFF-LINWDT(MNVLAB)-VCH/2-2 )
(0070)            IVL= IVL - VCH-2
(0071)            CALL MOVABS( IHL, IVL )
(0072)            CALL AOUTST( NVLAB2, VLABL2 )
(0073)            IF ( NVLAB3 .EQ. 0 ) GO TO 200
(0074)            IHL= MAX0( 0, HWL+VOFF-LINWDT(MNVLAB)-VCH/2-2 )
(0075)            IVL= IVL - VCH-2
(0076)            CALL MOVABS( IHL, IVL )
(0077)            CALL AOUTST( NVLAB3, VLABL3 )
(0078)      C
(0079)      C DECLARE OFFSETS AND DRAW LOWER AND LEFT SIDE AXES.
(0080)      C DSPLAY ALSO LABELS TIC MARKS.
(0081)      C
(0082)       200  CALL XLOC( HOFF )
(0083)            CALL YLOC( VOFF )
(0084)            CALL CHECK( 0, 0 )
(0085)            CALL DSPLAY( 0, 0 )
(0086)      C
(0087)      C DECLARE OFFSETS FOR TOP AND RIGHT SIDE AXES AND DRAW THEM.
(0088)      C
(0089)            CALL XLOCTP( -HOFF )
(0090)            CALL YLOCRT( -VOFF )
(0091)            CALL CHECK( 0, 0 )
(0092)            CALL GRID
(0093)      C
(0094)            RETURN
(0095)            END
     PROGRAM SIZE:   PROCEDURE - 000541    LINKAGE - 000124    STACK - 000102
     0000 ERRORS [<AXES  >FTN-REV15.3]
```

```
      SUBROUTINE GPBLK ( HPB, VPB )

(0096)        SUBROUTINE GPBLK ( HPB, VPB )
(0097)  C
(0098)  C PUTS PARAMETER BLOCK ON PLOT.
(0099)  C
(0100)  C-----------------------------------------------------------------
(0101)  C     HPB      (INPUT)   HORIZONTAL RASTER COORDINATE OF UPPER LEFT
(0102)  C                        CORNER.
(0103)                           INTEGER HPB
(0104)  C     VPB      (INPUT)   VERTICAL RASTER COORDINATE OF UPPER LEFT CORNER
(0105)                           INTEGER VPB
(0106)  C-----------------------------------------------------------------
(0106)  C COMMON BLOCKS                              (SEE INIT FOR DOCUMENTATION
(0106)  C-----------------------------------------------------------------
(0107)  C
(0108)        IHPB= HPB
(0109)        IVPB= VPB
(0110)  C
(0111)        DO 200 IPBL=1,NPBL
(0112)        IVPB= IVPB- LINHGT(1)- 2
(0113)        IF ( IVPB .GE. 0 ) GO TO 180
(0114)        IHPB= IHPB+ LINWDT(NPBCH+5)
(0115)        IVPB=  VPB- LINHGT(1)- 2
(0116)   180  CALL MOVABS( IHPB, IVPB )
(0117)        CALL AOUTST( NPBCH, PBLK(1,IPBL) )
(0118)   200  CONTINUE
(0119)  C
(0120)        RETURN
(0121)        END
PROGRAM SIZE:   PROCEDURE - 000132    LINKAGE - 000044    STACK - 000022
0000 ERRORS [<GPBLK >FTN-REV15.3]
```

SUBROUTINE GRHCON

```
(0001)            SUBROUTINE GRHCON
(0002)      C
(0003)      C CONTROLS GRAPHICAL OUTPUT.
(0004)      C
(0005)      C-----------------------------------------------------------------
(0005)      C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0005)      C-----------------------------------------------------------------
(0006)      C LOCAL DECLARATIONS
(0007)            REAL MSFL(2,1), MSFM)
(0008)            EQUIVALENCE (SCR1,MSFL)
(0009)            DIMENSION BLPR(1)
(0010)            EQUIVALENCE (SCR1,BLPR)
(0011)            INTEGER ESCALE
(0012)            DIMENSION PROFS(10)
(0013)            DIMENSION IRSP(1)
(0014)            INTEGER ALINE(1)
(0015)            EQUIVALENCE (SCR2,ALINE), (SCR2,IRSP)
(0016)            DATA PROFS / 0., .1, .5, .9, 1., 5*0. /, NPROFS /5/
(0017)      C
(0018)      C SET UP ALPHANUMERIC BLOCK OF PARAMETER VALUES FOR GRAPHICAL OUTPUT.
(0019)      C
(0020)            NPBL= 6
(0021)            NPBCH= 5* NCW
(0022)            ENCODE( NPBCH, 220, ALINE ) DXM
(0023)      220   FORMAT( 6HXL-XE=, F5.1, 3H CM, 6X )
(0024)            CALL SETHOL( PBLK(1,1), NPBCH, ALINE )
(0025)            ENCODE( NPBCH, 230, ALINE ) DYM
(0026)      230   FORMAT( 2HL=, F5.1, 3H CM, 10X )
(0027)            CALL SETHOL( PBLK(1,2), NPBCH, ALINE )
(0028)            ENCODE( NPBCH, 240, ALINE ) GAMMA
(0029)      240   FORMAT( 6HGAMMA=, 1PE9.2, 5X )
(0030)            CALL SETHOL( PBLK(1,3), NPBCH, ALINE )
(0031)            ENCODE( NPBCH, 250, ALINE ) DTDX
(0032)      250   FORMAT( 2HG=, F7.2, 9H DEG C/CM, 2X )
(0033)            CALL SETHOL( PBLK(1,4), NPBCH, ALINE )
(0034)            ENCODE( NPBCH, 260, ALINE ) DTDTM
(0035)      260   FORMAT( 4HEPS=, F7.3, 8H DEG/SEC, 1X )
(0036)            CALL SETHOL( PBLK(1,5), NPBCH, ALINE )
(0037)            ENCODE( NPBCH, 270, ALINE ) GFORCE
(0038)      270   FORMAT( 8HGRAVITY=, F6.2, 2H G, 4X )
(0039)            CALL SETHOL( PBLK(1,6), NPBCH, ALINE )
(0040)      C
(0041)      C GET PLOT SELECTION FROM OPERATOR.
(0042)      C
(0043)      500   CALL NEWFAG
(0044)            WRITE (CRT,FMT80) TITLE
(0045)            WRITE (CRT,510)
(0046)      510   FORMAT( // 21H      FUNCTION TO PLOT, 29X,
(0047)           *             14H      PLOT TYPE,
(0048)           *           // 36H  1   FINAL LOCAL AVERAGE COMPOSITION, 14X,
(0049)           *              22H  1   VERTICAL PROFILES,
(0050)           *           / 13H  2   VELOCITY, 37X,
(0051)           *              24H  2   HORIZONTAL PROFILES,
(0052)           *           / 19H  3   PRESSURE: P-P0, 31X,
(0053)           *              17H  3   VECTOR FIELD,
(0054)           *           / 34H  4   PRESSURE - BULK HYDROSTATIC P, 16X,
(0055)           *           / 20H  5   FRACTION LIQUID, 30X,
(0056)           *           / 14H  6   MASS FLOW, 36X,
(0057)           *           / 16H  7   SOLUTE FLOW, 34X)
(0058)      530   WRITE (CRT,535)
```

```
      SUBROUTINE GPHCON

 (0059)  535  FORMAT( /// 41HENTER ITEM NUMBER OF FUNCTION TO PLOT, OR,
 (0060)     *         /   19H       P TO PROCEED. )
 (0061)       IRSP(1)=4H
 (0062)       READ (CRT,FMTA4,ERR=540) IRSP
 (0063)       IF (IRSP(1).EQ.4HP    ) RETURN
 (0064)       IF (IRSP(1).EQ.4H     ) GO TO 540
 (0065)       DECODE ( NCW, *, IRSP, ERR=540 ) ITMP
 (0066)       IF ( ITMP.GT.0 .AND. ITMP.LT.8 ) GO TO 550
 (0067)  540  WRITE (CRT,FMT5)
 (0068)       GO TO 530
 (0069)  550  WRITE (CRT,555)
 (0070)  555  FORMAT( /   30HENTER ITEM NUMBER OF PLOT TYPE )
 (0071)       READ (CRT,*,ERR=560) ITMP
 (0072)       IF ( ITMP.GT.0 .AND. ITMP.LT.4 ) GO TO 570
 (0073)  560  WRITE (CRT,FMT5)
 (0074)       GO TO 550
 (0075)  C
 (0076)  570  GO TO (600,700,800,850,900,1000,1100), ITMP
 (0077)  590  FORMAT ( 23H***** INVALID SELECTION )
 (0078)  C
 (0079)  C PLOT FINAL COMPOSITION
 (0080)  C
 (0081)  600  GO TO ( 620, 610, 610 ), ITMP
 (0082)  610  WRITE (CRT,590)
 (0083)       GO TO 530
 (0084)  620  CMAX= -RINF
 (0085)       CMIN= RINF
 (0086)       DO 625 J=1,NJ
 (0087)       CMAX= AMAX1( CMAX, AVCS(J) )
 (0088)  625  CMIN= AMIN1( CMIN, AVCS(J) )
 (0089)       DIF= AMAX1( CMAX-CLO, CLO-CMIN, .2 )
 (0090)       CALL SCALE( CLO-DIF, CLO+DIF, SCMIN, SCMAX, NSI, ESCALE )
 (0091)  C
 (0092)       CALL NEWPAG
 (0093)  630  OMIN= SCMIN
 (0094)       OMAX= SCMAX
 (0095)       WRITE (CRT,635) CMIN,ALNAM(2), CMAX,ALNAM(2), SCMIN,SCMAX
 (0096)  635  FORMAT(    22HMINIMUM COMPOSITION IS, F6.2, 8H WT.PCT., A4,
 (0097)     *         /  22HMAXIMUM COMPOSITION IS, F6.2, 8H WT.PCT., A4,
 (0098)     *         ///37HAUTOMATIC SCALING YIELDS PLOT RANGE: , F6.2, 3H TO,
 (0099)     *           F6.2,
 (0100)     *         ///45HENTER P TO PROCEED WITH AUTOMATIC SCALING, OR,
 (0101)     *         /  35H       LOWER BOUND OF PLOT INTERVAL. )
 (0102)       DO 639 II=1,20
 (0103)  639  IRSP(II)=4H
 (0104)       READ (CRT,640,ERR=645) IRSP
 (0105)  640  FORMAT( 20A4 )
 (0106)       IF (IRSP(1).EQ.4HP    ) GO TO 695
 (0107)       IF (IRSP(1).EQ.4H     ) GO TO 645
 (0108)       DECODE( 80, *, IRSP, ERR=645 ) CMIN
 (0109)       GO TO 650
 (0110)  645  WRITE (CRT,FMT5)
 (0111)       GO TO 630
 (0112)  650  WRITE (CRT,655)
 (0113)  655  FORMAT(/  35HENTER UPPER BOUND OF PLOT INTERVAL. )
 (0114)       DO 659 II=1,20
 (0115)  659  IRSP(II)=4H
 (0116)       READ (CRT,640,ERR=665) IRSP
 (0117)       IF (IRSP(1).EQ.4H     ) GO TO 665
 (0118)       DECODE( 80, *, IRSP, ERR=665 ) OMAX
```

```
      SUBROUTINE GPMCON
(0119)      IF (CMAX.LE.CMIN) GO TO 645
(0120)      SCMIN= CMIN
(0121)      SCMAX= CMAX
(0122)      GO TO 695
(0123)  665 WRITE (CRT,FMT5)
(0124)      GO TO 650
(0125)  C
(0126)  695 CALL VPROFS( IDUM, DUM, SCMIN, SCMAX, NSI, -ESCALE,
(0127)     *             AVCS, 1, 1, JDIM, 0,
(0128)     *             32, 32HFINAL LOCAL AVERAGE COMPOSITION ,
(0129)     *             16, 16H (WT-PCT SOLUTE) )
(0130)      GO TO 500
(0131)  C
(0132)  C PLOT VELOCITY
(0133)  C
(0134)  700 GO TO ( 710, 710, 740 ), ITMP
(0135)  710 WRITE (CRT,590)
(0136)      GO TO 530
(0137)  740 VMAX= 0.
(0138)      DO 745 J=1,NJ
(0139)      DO 745 I=1,NI
(0140)  745 VMAX= AMAX1( VMAX, V(1,I,J)**2+V(2,I,J)**2 )
(0141)      VMAX= SQRT( VMAX )
(0142)      CALL VECPLT( VMAX, V, 2, IDIM, JDIM, 0,
(0143)     *             16, 16HVELOCITY FIELD , 8, 8H(CM/SEC) )
(0144)      GO TO 500
(0145)  C
(0146)  C PLOT PRESSURE (P-P0)
(0147)  C
(0148)  800 PMAX= -RINF
(0149)      PMIN= RINF
(0150)      DO 805 J=1,NJ
(0151)      DO 805 I=1,NI
(0152)      BLPR(I+(J-1)*NI)= P(I+1,J+1)+ RHL(NI,J)*GRAV*(Y(NJ)-Y(J))
(0153)      PMAX= AMAX1( PMAX, BLPR(I+(J-1)*NI) )
(0154)      PMIN= AMIN1( PMIN, BLPR(I+(J-1)*NI) )
(0155)  805 CONTINUE
(0156)      CALL SCALE ( PMIN, PMAX, SCMIN, SCMAX, NSI, ESCALE )
(0157)      GO TO ( 820, 840, 810 ), ITMP
(0158)  810 WRITE (CRT,590)
(0159)      GO TO 530
(0160)  820 CALL VPROFS( NPROFS, PROFS, SCMIN, SCMAX, NSI, -ESCALE,
(0161)     *             BLPR, 1, NI, NJ, 0,
(0162)     *             16, 16HPRESSURE: (P-P0),
(0163)     *             16, 16H (DYNES/CM**2) )
(0164)      GO TO 500
(0165)  840 CALL HPROFS( NPROFS, PROFS, SCMIN, SCMAX, NSI, -ESCALE,
(0166)     *             BLPR, 1, NI, NJ, 0,
(0167)     *             16, 16HPRESSURE: (P-P0),
(0168)     *             16, 16H(DYNES/CM**2)     )
(0169)      GO TO 500
(0170)  C
(0171)  C PLOT PRESSURE = BULK HYDROSTATIC PRESSURE.
(0172)  C
(0173)  850 PMAX= -RINF
(0174)      PMIN= RINF
(0175)      DO 855 J=1,NJ
(0176)      DO 855 I=1,NI
(0177)      PMAX= AMAX1( PMAX, P(I+1,J+1) )
(0178)      PMIN= AMIN1( PMIN, P(I+1,J+1) )
```

```
      SUBROUTINE GENCON
(0179)  855  CONTINUE
(0180)       CALL SCALE ( PMIN, PMAX, SCMIN, SCMAX, NSI, ESCALE )
(0181)       GO TO ( 870, 890, 860 ), ITMP
(0182)  860  WRITE (CRT,590)
(0183)       GO TO 530
(0184)  870  CALL VPROFS( NPROFS, PROFS, SCMIN, SCMAX, NSI, -ESCALE,
(0185)      *              P, 1, ICIM+2, JCIM+2, 1,
(0186)      *              32, 32HPRESSURE - BULK HYDROSTATIC P    ,
(0187)      *              16, 16H  (DYNES/CM**2) )
(0188)       GO TO 500
(0189)  890  CALL HPROFS( NPROFS, PROFS, SCMIN, SCMAX, NSI, -ESCALE,
(0190)      *              P, 1, ICIM+2, JCIM+2, 1,
(0191)      *              32, 32HPRESSURE - BULK HYDROSTATIC P    ,
(0192)      *              16, 16H(DYNES/CM**2)     )
(0193)       GO TO 500
(0194)  C
(0195)  C PLOT VOLUME FRACTION LIQUID.
(0196)  C
(0197)  900  GO TO ( 920, 940, 910 ), ITMP
(0198)  910  WRITE (CRT,590)
(0199)       GO TO 530
(0200)  920  CALL VPROFS( NPROFS, PROFS, 0., 1., 5, 0,
(0201)      *              GL, 1, IDIM, JDIM, 0,
(0202)      *              24, 24H VOLUME FRACTION LIQUID , 0, NULL )
(0203)       GO TO 500
(0204)  940  CALL HPROFS( NPROFS, PROFS, 0., 1., 5, 0,
(0205)      *              GL, 1, IDIM, JDIM, 0,
(0206)      *              24, 24H VOLUME FRACTION LIQUID , 0, NULL )
(0207)       GO TO 500
(0208)  C
(0209)  C MASS FLOW
(0210)  C
(0211)  1000 GO TO ( 1010, 1010, 1020 ), ITMP
(0212)  1010 WRITE (CRT,590)
(0213)       GO TO 530
(0214)  1020 MSFMX= 0.
(0215)       DO 1025 J=1,NJ
(0216)       DO 1025 I=1,NI
(0217)       IM=I+(J-1)*NI
(0218)       MSFL(1,IM)= RHL(I,J)*GL(I,J)*V(1,I,J)
(0219)       MSFL(2,IM)= RHL(I,J)*GL(I,J)*V(2,I,J)
(0220)       MSFMX= AMAX1( MSFMX, MSFL(1,IM)**2+MSFL(2,IM)**2 )
(0221)  1025 CONTINUE
(0222)       MSFMX= SQRT( MSFMX )
(0223)       CALL VECPLT( MSFMX, MSFL, 2, NI, NJ, 0,
(0224)      *              12, 12HMASS FLOW    ,
(0225)      *              20, 20H( GM/ (SEC* CM**2) ) )
(0226)       GO TO 500
(0227)  C
(0228)  C SOLUTE FLOW
(0229)  C
(0230)  1100 GO TO ( 1110, 1110, 1120 ), ITMP
(0231)  1110 WRITE (CRT,590)
(0232)       GO TO 530
(0233)  1120 MSFMX= 0.
(0234)       DO 1125 J=1,NJ
(0235)       DO 1125 I=1,NI
(0236)       IM=I+(J-1)*NI
(0237)       MSFL(1,IM)= .01* CL(I,J)* RHL(I,J)* GL(I,J)* V(1,I,J)
(0238)       MSFL(2,IM)= .01* CL(I,J)* RHL(I,J)* CL(I,J)* V(2,I,J)
```

```
      SUBROUTINE GPHCON
(0239)       MSFMX= AMAX1( MSFMX, MSFL(1,IM1)**2+MSFL(2,IM)**2 )
(0240)  1125 CONTINUE
(0241)       MSFMX=SQRT(MSFMX)
(0242)       CALL VECPLT( MSFMX, MSFL, 2, NI, NJ, 0,
(0243)      *             12,12HSOLUTE FLOW ,
(0244)      *             20, 20H( GM/ (SEC* CM**2) ) )
(0245)       GO TO 500
(0246)  C
(0247)       END
 PROGRAM SIZE:   PROCEDURE - 004413    LINKAGE - 000327    STACK - 000032
 3000 ERRORS [<GPHCON>FTN-REV15.3]
```

```
      SUBROUTINE HPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,
```

```
(0336)        SUBROUTINE HPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,
(0337)       *                    ARRAY, N1, N2, N3, IJE, NC, NAME,
(0338)       *                    NCU, NAMEU )
(0339)  C
(0340)  C PLOTS SELECTED HORIZONTAL PROFILES OF ARRAY.
(0341)  C
(0342)  C-----------------------------------------------------------------------
(0343)  C     NPRF       (INPUT)    NUMBER OF PROFILES TO PLOT.
(0344)  C     PRF        (INPUT)    YY VALUES OF PROFILES TO PLOT.
(0345)  C                           WILL INTERPOLATE LINEARLY TO VALUES BETWEEN
(0346)  C                           TABULATED VALUES OF YY.
(0347)                              DIMENSION PRF(NPRF)
(0348)  C     AMIN       (INPUT)    MINIMUM VALUE ON HORIZONTAL AXIS (SCALED
(0349)  C                           BY 10**ASCALE).
(0350)  C     AMAX       (INPUT)    MAXIMUM VALUE ON HORIZONTAL AXIS (SCALED
(0351)  C                           BY 10**ASCALE).
(0352)  C     NATCS      (INPUT)    NUMBER OF TIC SPACES ON HORIZONTAL AXIS.
(0353)  C     ASCALE     (INPUT)    SCALE FACTOR FOR PLOTTING WILL BE
(0354)  C                           10**ASCALE.
(0355)                              INTEGER ASCALE
(0356)  C     ARRAY      (INPUT)    ARRAY TO PLOT.
(0357)                              DIMENSION ARRAY(N1,N2,N3)
(0358)  C     N1,N2,N3   (INPUT)    DIMENSIONS OF ARRAY.
(0359)  C     IJE        (INPUT)    EXPANDED MESH INDICATOR.  SEE PSOLVE.
(0360)  C     NC         (INPUT)    NUMBER OF CHARACTERS IN NAME OF ARRAY.
(0361)  C     NAME       (INPUT)    NAME OF ARRAY -USED TO LABEL VERTICAL AXIS.
(0362)                              DIMENSION NAME(NC)
(0363)  C     NCU        (INPUT)    NUMBER OF CHARACTERS IN ARRAY UNITS.
(0364)  C     NAMEU      (INPUT)    ARRAY UNITS - USED TO LABEL VERTICAL AXIS.
(0365)                              DIMENSION NAMEU(NCU)
(0366)  C-----------------------------------------------------------------------
(0366)  C COMMON BLOCKS                                    (SEE INIT FOR DOCUMENTATION
(0366)  C-----------------------------------------------------------------------
(0367)  C LOCAL DECLARATIONS
(0368)        DIMENSION APRF(1)
(0369)        EQUIVALENCE (SCR2,APRF)
(0370)        DIMENSION LABEL(1)
(0371)        EQUIVALENCE (SCR2,LABEL)
(0372)        DIMENSION LABSCL(3)
(0373)        INTEGER HK, VK, HPB, VPB
(0374)        INTEGER PAT(6)
(0375)        DATA PAT / 11, 56, 34, 12, 3212, 321212 /
(0376)  C-----------------------------------------------------------------------
(0377)  C
(0378)  C INITIALIZE PLOT, PUT ON LABELS AND TITLES, AND DRAW AXES.
(0379)  C LABEL WILL CONTAIN THE NAME OF ARRAY, THE SCALE FACTOR, IF
(0380)  C ANY, AND THE UNITS OF ARRAY.
(0381)  C GPBLK PUTS ON THE PARAMETER BLOCK.
(0382)  C
(0383)        CALL SETPLT( 0, 24, 24H HORIZONTAL PROFILES OF , NC, NAME    )
(0384)  C
(0385)        HPB= HWR+VCH+HCH
(0386)        VPB= VWU- (VL-NPBL*(VCH+2))/2
(0387)        CALL GPBLK( HPB, VPB )
(0388)  C
(0389)        NCL= 12
(0390)        ENCODE ( NCL, 210, LABSCL ) ASCALE
(0391)        IF ( ASCALE.EQ.0 ) NCL= 0
(0392)   210  FORMAT ( 7H X 10**, I3, 2X )
(0393)  C
```

```
      SUBROUTINE HPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,

(0394)       CALL AXFS( 0.,   1.,   0, 5,-  --2, 16, 16H-(X-XE)/(XL-XE) ,
(0395)      *           AMIN, AMAX, 0, NATCS, 2, NC, NAME, NCL, LABSCL,
(0396)      *           NCU, NAMEU )
(0397)  C
(0398)  C SET PLOT VARIABLES FOR CURVES.
(0399)  C
(0400)       CALL NPTS( NI )
(0401)       CALL DWINDO( 0., 1., AMIN/(10.**ASCALE), AMAX/(10.**ASCALE) )
(0402)  C
(0403)       IF (NC.EQ.1) GO TO 500
(0404)  C
(0405)  C INITIALIZE OF LINE PATTERN KEY BLOCK.
(0406)  C
(0407)       LKL= 50
(0408)       LKT= LKL+LINWDT(12)
(0409)       VK= VWL-3*VCH-3-2*VCH
(0410)       NKB= NPRF/ (VK/(VCH+2))
(0411)       IF ( (VK/(VCH+2))*NKB .LT. NPRF ) NKB= NKB+1
(0412)       HK= LINWDT( MAX0(24,NC) + 5 )
(0413)       HK= MAX0( HK, HSCRN- NKB*LKT- (NKB-1)*LINWDT(5) )
(0414)       IHK= HK
(0415)       IVK= VK+VCH+2
(0416)  C
(0417)  C LOOP THROUGH PROFILES.
(0418)  C
(0419)       DO 480 IPRF=1,NPRF
(0420)  C
(0421)  C       LOCATE EACH PRF IN YY, INTERPOLATE TO PRF, AND PLOT CURVE.
(0422)  C
(0423)          DO 320 J=2,NJ
(0424)   320    IF ( PRF(IPRF) .LT. YY(J) ) GO TO 340
(0425)          J=NJ
(0426)  C
(0427)   340    FACT= (PRF(IPRF)-YY(J-1)) / (YY(J)-YY(J-1))
(0428)          DO 360 I=1,NI
(0429)          APRF(I)= ARRAY(1,I+IJE,J+IJE-1)+
(0430)      *              FACT*( ARRAY(1,I+IJE,J+IJE)-ARRAY(1,I+IJE,J+IJE-1) )
(0431)   360    CONTINUE
(0432)  C
(0433)          IPAT= PAT( MOD(IPRF+5,6)+1 )
(0434)          CALL LINE( IPAT )
(0435)          CALL CPLOT( XX, APRF )
(0436)  C
(0437)  C       LABEL LINE PATTERN
(0438)  C
(0439)          IVK= IVK-VCH-2
(0440)          IF (IVK.GE.0) GO TO 420
(0441)          IVK= VK
(0442)          IHK= IHK+LKT+LINWDT(5)
(0443)   420    IF ( IHK+LKT .GT. HSCRN ) GO TO 480
(0444)          CALL MOVABS ( IHK, IVK+(VCH+2)/3 )
(0445)          CALL DSHABS( IHK+LKL, IVK+(VCH+2)/3, IPAT )
(0446)          CALL MOVABS( IHK+LKL, IVK )
(0447)          IF (IVK.EQ.VK) ENCODE( 12, 430, LABEL ) PRF(IPRF)
(0448)          IF (IVK.LT.VK) ENCODE( 12, 440, LABEL ) PRF(IPRF)
(0449)   430    FORMAT( 6H Y/L =, F6.3 )
(0450)   440    FORMAT( 6X,       F6.3 )
(0451)          CALL AOUTST( 12, LABEL )
(0452)  C
(0453)   480  CONTINUE
```

```
SUBROUTINE HPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,

(0454)  C
(0455)        GO TO 600
(0456)  C
(0457)  C PLOT SINGLY DIMENSIONED ARRAY.
(0458)  C
(0459)   500  CALL CPLOT( XX, ARRAY )
(0460)  C
(0461)  C FLUSH PLOT BUFFER, RETURN TO ALPHANUMERIC MODE, AND WAIT FOR OPERATOR
(0462)  C RESPONSE.
(0463)  C
(0464)   600  CALL MOVABS( 0, VSCRN )
(0465)        CALL ANMODE
(0466)        CALL WAIT
(0467)        RETURN
(0468)  C
(0469)        END
PROGRAM SIZE:   PROCEDURE - 001263   LINKAGE - 000212   STACK - 000102
0000 ERRORS [<HPROFS>FTN-REV15.3]
```

```
      SUBROUTINE SCALE ( FMIN, FMAX, SCMIN, SCMAX, NSI, ESCALE )
```

```
(0470)       SUBROUTINE SCALE ( FMIN, FMAX, SCMIN, SCMAX, NSI, ESCALE )
(0471) C
(0472) C CALCULATE APPROPRIATE PLOT SCALE FOR FUNCTION.
(0473) C
(0474) C-----------------------------------------------------------------------
(0475) C     FMIN     (INPUT)    MINIMUM OF FUNCTION TO SCALE.
(0476) C     FMAX     (INPUT)    MAXIMUM OF FUNCTION TO SCALE.
(0477) C     SCMIN    (OUTPUT)   MINIMUM SCALE VALUE.
(0478) C     SCMAX    (OUTPUT)   MAXIMUM SCALE VALUE.
(0479) C     NSI      (OUTPUT)   NUMBER OF TIC INTERVALS ON SCALE BETWEEN
(0480) C                         SCMAX AND SCMIN.
(0481) C     ESCALE   (OUTPUT)   SCALE VALUES MULTIPLIED BY 10**-ESCALE.
(0482)                           INTEGER ESCALE
(0483) C-----------------------------------------------------------------------
(0484) C LOCAL DECLARATIONS
(0485)       INTEGER EX
(0486) C-----------------------------------------------------------------------
(0487) C
(0488) C DETERMINE REMOTE EXPONENT, IF ANY.
(0489) C
(0490)       SCMIN=0.
(0491)       SCMAX= 0.
(0492)       NSI= 0
(0493)       ESCALE= 0
(0494)       AMAXF= AMAX1( ABS(FMIN), ABS(FMAX) )
(0495)       IF (AMAXF.EQ.0.) RETURN
(0496)       ESCALE= IFIX( ALOG10(AMAXF) )
(0497)       IF ( ALOG10(AMAXF)-FLOAT(ESCALE) .GT. .999 ) ESCALE= ESCALE+1
(0498)       IF ( IABS(ESCALE) .LE. 2 ) ESCALE= 0
(0499) C
(0500) C DETERMINE TIC INTERVAL.
(0501) C
(0502)       ADIFF= ABS( FMAX-FMIN )
(0503)       IF (ADIFF.EQ.0.) RETURN
(0504)       FLOG= ALOG10( ADIFF )
(0505)       EX= IFIX( FLOG )
(0506)       IF ( FLOG .LT. 0. ) EX= EX-1
(0507)       WH= ADIFF* 10.**(-EX)
(0508)       DEL=1.
(0509)       IF ( WH .LT. 4.5 ) DEL= .5
(0510)       IF ( WH .LT. 2.  ) DEL= .2
(0511)       IF ( WH .LT. 1.2 ) DEL= .1
(0512) C
(0513) C DETERMINE SCALE MAXIMUM AND MINIMUM.
(0514) C AVOID CLIPPING AT EXTREMES.
(0515) C
(0516)       SMIN= FMIN* 10.**(-EX)
(0517)       NMIN= IABS( IFIX(SMIN/DEL) )
(0518)       IF ( ( SMIN .GT. 0. ) .AND.
(0519)      *     ( SMIN-FLOAT(NMIN)*DEL .GT. .99*DEL ) ) NMIN= NMIN+1
(0520)       IF ( ( SMIN .LT. 0. ) .AND.
(0521)      *     (-SMIN-FLOAT(NMIN)*DEL .GT. .01*DEL ) ) NMIN= NMIN+1
(0522)       SMAX= FMAX* 10.**(-EX)
(0523)       NMAX= IABS( IFIX(SMAX/DEL) )
(0524)       IF ( ( SMAX .GT. 0. ) .AND.
(0525)      *     ( SMAX-FLOAT(NMAX)*DEL .GT. .01*DEL ) ) NMAX= NMAX+1
(0526)       IF ( ( SMAX .LT. 0. ) .AND.
(0527)      *     (-SMAX-FLOAT(NMAX)*DEL .GT. .99*DEL ) ) NMAX= NMAX+1
(0528)       SCMIN= SIGN(1.,FMIN)* FLOAT(NMIN)* DEL* 10.**(EX-ESCALE)
(0529)       SCMAX= SIGN(1.,FMAX)* FLOAT(NMAX)* DEL* 10.**(EX-ESCALE)
```

```
      SUBROUTINE SCALE ( FMIN, FMAX, SCMIN, SCMAX, NSI, ESCALE )

(0530)        NSI= IABS( NMAX* IFIX( -SIGN(1.,FMIN*FMAX) )* NMIN )
(0531)  C
(0532)        RETURN
(0533)        END
PROGRAM SIZE:   PROCEDURE - 000600    LINKAGE - 000056    STACK - 000046
0000 ERRORS [<SCALE >FTN-REV15.3]
```

SUBROUTINE SETPLT (HOFF, NCLAB1, LABEL1, NCLAB2, LABEL2)

```
(0534)       SUBROUTINE SETPLT ( HOFF, NCLAB1, LABEL1, NCLAB2, LABEL2 )
(0535)  C
(0536)  C INITIALIZE PLOT AND PUT ON PLOT LABEL.
(0537)  C
(0538)  C-------------------------------------------------------------------
(0539)  C     HOFF      (INPUT)    VERTICAL OFFSET OF HORIZONTAL AXIS FROM VWL.
(0540)  C                          RASTER UNITS.
(0541)                             INTEGER HOFF
(0542)  C     NCLAB1    (INPUT)    NUMBER OF CHARACTERS IN PLOT LABEL 1.
(0543)  C     LABEL1    (INPUT)    PLOT LABEL 1 (ALPHANUMERIC).
(0544)                             DIMENSION LABEL1(NCLAB1)
(0545)  C     NCLAB2    (INPUT)    NUMBER OF CHARACTERS IN PLOT LABEL 2.
(0546)  C     LABEL2    (INPUT)    PLOT LABEL 2 (ALPHANUMERIC).
(0547)                             DIMENSION LABEL2(NCLAB2)
(0548)  C-------------------------------------------------------------------
(0548)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATION
(0548)  C-------------------------------------------------------------------
(0549)  C
(0550)  C CLEAR SCREEN AND INITIALIZE PLOT COMMON AREA
(0551)  C
(0552)       CALL NEWPAG
(0553)       CALL BINITT
(0554)  C
(0555)  C GET CHARACTER SIZES.
(0556)  C
(0557)       CALL CSIZE( HCH, VCH )
(0558)  C
(0559)  C SET UP VIRTUAL WINDOW.  THE SCREEN SIZE IS HSCRN BY VSCRN RASTER
(0560)  C UNITS.  THE NOMINAL PLOT WINDOW IS HW BY VW RASTER UNITS.  TIC MARKS
(0561)  C WILL EXTEND OUTSIDE THE WINDOW A DISTANCE VCH.
(0562)  C
(0563)       VWU= VSCRN- 3*VCH- 2+ HOFF
(0564)       VWL= VWU-VW+1
(0565)       HWL= (HSCRN-HW)/2 -3*HCH
(0566)       HWR= HWL+HW-1
(0567)       CALL SLIMX( HWL, HWR )
(0568)       CALL SLIMY( VWL, VWU )
(0569)  C
(0570)  C CENTER TITLE BLOCK ACROSS TOP OF VIRTUAL WINDOW.
(0571)  C
(0572)       IHL= HWL + (HW-LINWDT(80))/2
(0573)       IVL= VSCRN-VCH
(0574)       CALL MOVABS( IHL, IVL )
(0575)       CALL AOUTST( 80, TITLE(1,1) )
(0576)       IVL= IVL-VCH-1
(0577)       CALL MOVABS( IHL, IVL )
(0578)       CALL AOUTST( 80, TITLE(1,2) )
(0579)  C
(0580)  C PUT PLOT LABELS IN LOWER LEFT CORNER.
(0581)  C
(0582)       IHL= 0
(0583)       IVL= VCH+2
(0584)       CALL MOVABS( IHL, IVL )
(0585)       CALL AOUTST( NCLAB1, LABEL1 )
(0586)       IHL= MAX0( 0, (NCLAB1-NCLAB2)*HCH/2 )
(0587)       IVL= 0
(0588)       CALL MOVABS( IHL, IVL )
(0589)       CALL AOUTST( NCLAB2, LABEL2 )
(0590)  C
(0591)       RETURN
```

```
      SUBROUTINE SETPLT ( HOFF, NCLAB1, LABEL1, NCLAB2, LABEL2 )

(0592)       END
PROGRAM SIZE:   PROCEDURE - 000266    LINKAGE - 000076    STACK - 000034
0000 ERRORS [<SETPLT>FTN-REV15.3]
```

```
      SUBROUTINE VECPLT ( VSCALE,
```

```
(0593)        SUBROUTINE VECPLT ( VSCALE,
(0594)       *                    ARRAY, N1, N2, N3, IJE, NC, NAME, NCU, NAMEU
(0595)  C
(0596)  C DOES VECTOR PLOTS OF ARRAY.
(0597)  C
(0598)  C-----------------------------------------------------------------------
(0599)  C     VSCALE     (INPUT)    PLOT SCALE WILL BE 1 CM = VSCALE UNITS.
(0600)  C     ARRAY      (INPUT)    ARRAY OF VECTORS TO PLOT.
(0601)                              DIMENSION ARRAY(N1,N2,N3)
(0602)  C     N1,N2,N3   (INPUT)    DIMENSIONS OF ARRAY.  (N1 .GE. 2)
(0603)  C     IJE        (INPUT)    EXTENDED MESH INDICATOR.  SEE PSOLVE.
(0604)  C     NC         (INPUT)    NUMBER OF CHARACTERS IN NAME.
(0605)  C     NAME       (INPUT)    NAME OF ARRAY.
(0606)                              DIMENSION NAME(NC)
(0607)  C     NCU        (INPUT)    NUMBER OF CHARACTERS IN UNITS LABEL.
(0608)  C     NAMEU      (INPUT)    UNITS LABEL.
(0609)                              DIMENSION NAMEU(NCU)
(0610)  C-----------------------------------------------------------------------
(0610)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATIO
(0610)  C-----------------------------------------------------------------------
(0611)  C LOCAL DECLARATIONS
(0612)        INTEGER OFFSET, HPB, VPB
(0613)        INTEGER LABEL(1), LSCALE(3), BLANK, EQSIGN
(0614)        EQUIVALENCE (SCR2,LABEL)
(0615)        DATA BLANK /4H    /, EQSIGN /4H  =/
(0616)        DATA SCALE / 48. /
(0617)  C-----------------------------------------------------------------------
(0618)  C
(0619)  C INITIALIZE PLOT, DRAW AXES, AND PUT ON TITLE.
(0620)  C
(0621)        NCS= IFIX(SCALE)/HCH
(0622)        NWS= NCS/NCW
(0623)        ENCODE( 12, 210, LSCALE ) VSCALE
(0624)   210  FORMAT( 1PE10.3, 2X )
(0625)        NNW= NCU/NCW
(0626)        IF ( NNW*NCW .LT. NCU ) NNW= NNW+1
(0627)        NCL= 8+ NWS*NCW+ 4+ 12+ NNW*NCW
(0628)        ENCODE( NCL, 220, LABEL ) (BLANK,IWS=1,NWS), EQSIGN, LSCALE,
(0629)       *                          (NAMEU(INW),INW=1,NNW)
(0630)   220  FORMAT( 8HSCALE:  , 40(A4) )
(0631)  C
(0632)        OFFSET= -IFIX(SCALE/2.)
(0633)        CALL SETPLT( OFFSET, NC, NAME, NCL, LABEL )
(0634)        CALL MOVABS( LINWDT(7), (VCH+2)/3 )
(0635)        CALL DRWABS( LINWDT(7)+IFIX(SCALE), (VCH+2)/3 )
(0636)  C
(0637)        CALL AXES( 0., 1., OFFSET, 5, 2, 10, 16H (X-XE)/(XL-XE) ,
(0638)       *           0., 1., OFFSET, 5, 2, 8,  8H  Y/L   ,
(0639)       *           0, NULL, 0, NULL )
(0640)  C
(0641)  C PUT PARAMETER BLOCK ON PLOT.
(0642)  C
(0643)        VPB= VWL+ OFFSET- 3*VCH-3- 2*VCH
(0644)        NPB= NPBL/ (VPB/(VCH+2))
(0645)        IF ( (VPB/(VCH+2))*NPB .LT. NPBL ) NPB= NPB+1
(0646)        HPB= LINWDT( MAX0(NC,NCL) + 5 )
(0647)        HPB= MAX0( HPB, HSCRN- NPB*LINWDT(NPBCH)- (NPB-1)*LINWDT(5) )
(0648)        CALL GPBLK( HPB, VPB )
(0649)  C
(0650)  C
```

```
      SUBROUTINE VECPLT ( VSCALE,
C INITIALIZE ARROWHEAD CALCULATION AND BEGIN LOOPS THROUGH MESH.
C
      HANG= 30.* ATAN2(1.,0.)/ 90.
      HR= 1./COS(HANG)
      HBASE= 2.* HR* SIN(HANG)
C
      DO 500 J=1,NJ
      DO 500 I=1,NI
C
C        DRAW VECTOR BODY.
C
         XVEC= XX(I)*FLOAT(HWL)
         YVEC= YY(J)*FLOAT(VWL)
         CALL MOVABS( HWL*IFIX(XVEC), VWL*IFIX(YVEC) )
         XVEC= SCALE* ARRAY(1,I+IJE,J+IJE)/VSCALE + XVEC
         YVEC= SCALE* ARRAY(2,I+IJE,J+IJE)/VSCALE + YVEC
         CALL DRWABS( HWL*IFIX(XVEC), VWL*IFIX(YVEC) )
C
C        DRAW VECTOR HEAD.
C        NOMINAL HEAD LENGTH IS .15* VECTOR LENGTH.
C        MINIMUM HEAD LENGTH IS .1* VSCALE.
C
         VLNTH= SQRT( ARRAY(1,I+IJE,J+IJE)**2 +
     *                ARRAY(2,I+IJE,J+IJE)**2 ) / VSCALE
         IF ( VLNTH .LT. PRCSN ) GO TO 500
         VANG= ATAN2( ARRAY(2,I+IJE,J+IJE), ARRAY(1,I+IJE,J+IJE) )
         HL= AMAX1( .15* VLNTH, .1 )
         XVECH= SCALE* HL*HR* COS(VANG+HANG)
         YVECH= SCALE* HL*HR* SIN(VANG+HANG)
         CALL DRWABS( HWL*IFIX(XVEC-XVECH), VWL*IFIX(YVEC-YVECH) )
         XVECH=  SCALE* HL*HBASE* SIN(VANG) + XVECH
         YVECH= -SCALE* HL*HBASE* COS(VANG) + YVECH
         CALL MOVABS( HWL*IFIX(XVEC), VWL*IFIX(YVEC) )
         CALL DRWABS( HWL*IFIX(XVEC-XVECH), VWL*IFIX(YVEC-YVECH) )
C
  500  CONTINUE
C
C FLUSH PLOT BUFFER, RETURN TO ALPHANUMERIC MODE, AND WAIT FOR OPERATI
C RESPONSE.
C
  600  CALL MOVABS( 0, VSCRN )
       CALL ANMODE
       CALL WAIT
       RETURN
C
       END
PROGRAM SIZE:   PROCEDURE - 001403    LINKAGE - 000223    STACK - 000066
0000 ERRORS [<VECPLT>FTN-REV15.3]
```

```
SUBROUTINE VPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,

(0697)        SUBROUTINE VPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,
(0698)       *                    ARRAY, N1, N2, N3, IJE, NC, NAME,
(0699)       *                    NCU, NAMEU )
(0700)  C
(0701)  C PLOTS SELECTED VERTICAL PROFILES OF ARRAY.
(0702)  C
(0703)  C-----------------------------------------------------------------
(0704)  C     NPRF       (INPUT)    NUMBER OF PROFILES TO PLOT.
(0705)  C     PRF        (INPUT)    XX VALUES OF PROFILES TO PLOT.
(0706)  C                           WILL INTERPOLATE LINEARLY TO VALUES BETWEEN
(0707)  C                           TABULATED VALUES OF XX.
(0708)                              DIMENSION PRF(NPRF)
(0709)  C     AMIN       (INPUT)    MINIMUM VALUE ON VERTICAL AXIS (SCALED
(0710)  C                           BY 10**ASCALE).
(0711)  C     AMAX       (INPUT)    MAXIMUM VALUE ON VERTICAL AXIS (SCALED
(0712)  C                           BY 10**ASCALE).
(0713)  C     NATCS      (INPUT)    NUMBER OF TIC SPACES ON VERTICAL AXIS.
(0714)  C     ASCALE     (INPUT)    SCALE FACTOR FOR PLOTTING WILL BE
(0715)  C                           10**ASCALE.
(0716)                              INTEGER ASCALE
(0717)  C     ARRAY      (INPUT)    ARRAY TO PLOT.
(0718)                              DIMENSION ARRAY(N1,N2,N3)
(0719)  C     N1,N2,N3   (INPUT)    DIMENSIONS OF ARRAY.
(0720)  C     IJE        (INPUT)    EXPANDED MESH INDICATOR.  SEE FSOLVE.
(0721)  C     NC         (INPUT)    NUMBER OF CHARACTERS IN NAME OF ARRAY.
(0722)  C     NAME       (INPUT)    NAME OF ARRAY -USED TO LABEL HORIZONTAL AXI
(0723)                              DIMENSION NAME(NC)
(0724)  C     NCU        (INPUT)    NUMBER OF CHARACTERS IN ARRAY UNITS.
(0725)  C     NAMEU      (INPUT)    ARRAY UNITS - USED TO LABEL HORIZONTAL AXIS
(0726)                              DIMENSION NAMEU(NCU)
(0727)  C-----------------------------------------------------------------
(0727)  C COMMON BLOCKS                                  (SEE INIT FOR DOCUMENTATI(
(0727)  C-----------------------------------------------------------------
(0728)  C LOCAL DECLARATIONS
(0729)        DIMENSION APRF(1)
(0730)        EQUIVALENCE (SCR2,APRF)
(0731)        DIMENSION LABEL(1)
(0732)        EQUIVALENCE (SCR2,LABEL)
(0733)        DIMENSION LABSCL(3)
(0734)        INTEGER HK, VK, HPB, VPB
(0735)        INTEGER PAT(6)
(0736)        DATA PAT / 11, 56, 34, 12, 3212, 321212 /
(0737)  C-----------------------------------------------------------------
(0738)  C
(0739)  C INITIALIZE PLOT, PUT ON LABELS AND TITLES, AND DRAW AXES.
(0740)  C LABEL WILL CONTAIN THE NAME OF ARRAY, THE SCALE FACTOR, IF
(0741)  C ANY, AND THE UNITS OF ARRAY.
(0742)  C GPBLK PUTS ON THE PARAMETER BLOCK.
(0743)  C
(0744)        CALL SETPLT( 0, 20, 20HVERTICAL PROFILES OF, NC, NAME )
(0745)  C
(0746)        HPB= HWR+VCH+HCH
(0747)        VPB= VWU- (VW-NPRL*(VCH+2))/2
(0748)        CALL GPBLK( HPB, VPB )
(0749)  C
(0750)        NNW= NC/NCW
(0751)        IF ( NCW*NNW .LT. NC ) NNW=NNW+1
(0752)        ENCODE ( 12, 210, LABSCL ) ASCALE
(0753)   210  FORMAT ( 7H X 10**, I3, 2X )
(0754)        DO 215 INW=1,3
```

```
      SUBROUTINE VPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,
(0755)    215  LABEL(INW+NNW)= LABSCL(INW)
(0756)         DO 230 INW=1,NNW
(0757)    230  LABEL(INL)=NAME(INW)
(0758)         NNL= NNW+3
(0759)         IF (ASCALE.EQ.0) NNL= NNW
(0760)         NNW= NCU/NCW
(0761)         IF ( NCW*NNW .LT. NCU ) NNW= NNW+1
(0762)         DO 240 INW=1,NNW
(0763)    240  LABEL(INW+NNL)= NAMEU(INW)
(0764)         NCL= (NNL+NNW)*NCW
(0765)    C
(0766)         CALL AXES( AMIN, AMAX, 0, NATCS, 2, NCL, LABEL,
(0767)        *           0.,   1.,   0, 5,    -2, .8,  8H  Y/L    ,
(0768)        *           0, NULL, 0, NULL )
(0769)    C
(0770)    C SET PLOT VARIABLES FOR CURVES.
(0771)    C
(0772)         CALL NPTS( NJ )
(0773)         CALL DWINDO( AMIN/(10.**ASCALE), AMAX/(10.**ASCALE), 0., 1. )
(0774)    C
(0775)         IF (N2.EQ.1) GO TO 500
(0776)    C
(0777)    C INITIALIZE OF LINE PATTERN KEY BLOCK.
(0778)    C
(0779)         LKL= 50
(0780)         LKT= LKL+LINWDT(24)
(0781)         VK= VWL-3*VCH-3-2*VCH
(0782)         NKB= NPRF/ (VK/(VCH+2))
(0783)         IF ( (VK/(VCH+2))*NKB .LT. NPRF ) NKB= NKB+1
(0784)         HK= LINWDT( MAXO(20,NC) + 5 )
(0785)         HK= MAXO( HK, HSCRN- NKB*LKT- (NKB-1)*LINWDT(5) )
(0786)         IHK= HK
(0787)         IVK= VK+VCH+2
(0788)    C
(0789)    C LOOP THROUGH PROFILES.
(0790)    C
(0791)         DO 480 IPRF=1,NPRF
(0792)    C
(0793)    C        LOCATE EACH PRF IN XX, INTERPOLATE TO PRF, AND PLOT CURVE.
(0794)    C
(0795)           DO 320 I=2,NI
(0796)     320   IF ( PRF(IPRF) .LT. XX(I) ) GO TO 340
(0797)           I=NI
(0798)    C
(0799)     340   FACT= (PRF(IPRF)-XX(I-1)) / (XX(I)-XX(I-1))
(0800)           DO 360 J=1,NJ
(0801)           APRF(J)= ARRAY(1,I+IJE-1,J+IJE)+
(0802)          *          FACT*( ARRAY(1,I+IJE,J+IJE)-ARRAY(1,I+IJE-1,J+IJE) )
(0803)     360   CONTINUE
(0804)    C
(0805)           IPAT= PAT( MOD(IPRF+5,6)+1 )
(0806)           CALL LINE( IPAT )
(0807)           CALL CPLOT( APRF, YY )
(0808)    C
(0809)    C        LABEL LINE PATTERN
(0810)    C
(0811)           IVK= IVK-VCH-2
(0812)           IF (IVK.GE.0) GO TO 420
(0813)           IVK= VK
(0814)           IHK= IHK+LKT+LINWDT(5)
```

```
      SUBROUTINE VPROFS ( NPRF, PRF, AMIN, AMAX, NATCS, ASCALE,

(0815)   420      IF ( IHK+LKT .GT. HSCRN ) GO TO 480
(0816)            CALL MOVABS ( IHK, IVK+(VCH+2)/3 )
(0817)            CALL DSHABS( IHK+LKL, IVK+(VCH+2)/3, IPAT )
(0818)            CALL MOVABS( IHK+LKL, IVK )
(0819)            IF (IVK.EQ.VK) ENCODE( 24, 430, LABEL ) PRF(IPRF)
(0820)            IF (IVK.LT.VK) ENCODE( 24, 440, LABEL ) PRF(IPRF)
(0821)   430      FORMAT( 18H  (X-XE)/(XL-XE) =, F6.3 )
(0822)   440      FORMAT( 18X,                   F6.3 )
(0823)            CALL AOUTST( 24, LABEL )
(0824)   C
(0825)    480  CONTINUE
(0826)   C
(0827)         GO TO 600
(0828)   C
(0829)   C PLOT SINGLY DIMENSIONED ARRAY.
(0830)   C
(0831)    500  CALL CPLOT( ARRAY, YY )
(0832)   C
(0833)   C FLUSH PLOT BUFFER, RETURN TO ALPHANUMERIC MODE, AND WAIT FOR OPERATOR
(0834)   C RESPONSE.
(0835)   C
(0836)    600  CALL MOVABS( 0, VSCRN )
(0837)         CALL ANMODE
(0838)         CALL WAIT
(0839)         RETURN
(0840)   C
(0841)         END
PROGRAM SIZE:   PROCEDURE - 001466     LINKAGE - 000227     STACK - 000102
0000 ERRORS [<VPROFS>FTN-REV15.3]
```

APPENDIX B

DATA BASE MANAGER LISTING

```
(0001)  C
(0002)  C-----------------------------------------------------------------
(0003)  C
(0004)  C CONTROLLER FOR ALLOY DATA BASE MANAGER.
(0005)  C
(0006)  C MODEL 1 (12/79)
(0007)  C
(0008)  C            * SETS UP CARD IMAGE DATA BASE FOR INPUT TO
(0009)  C              THE MPS SOLIDIFICATION MODEL.
(0010)  C
(0011)  C-----------------------------------------------------------------
(0012)  C COMMON BLOCKS
(0013)        COMMON       CRT, OLDDB, NEWDB,
(0014)       *             NAM(13,15), NN, VLU(3,15), NV,
(0015)       *             SOLVNT, SOLUTE, ALREF(20,5),
(0016)       *             FERROR(7), FBOUND(14), FA4(1), FBOCH(2),
(0017)       *             FMT1(5), FMT2(2), GUIDE(20)
(0018)        INTEGER      CRT, OLDDB, NEWDB,
(0019)       *             SOLVNT, SOLUTE, ALREF,
(0020)       *             FERROR, FBOUND, FA4, FBOCH,
(0021)       *             FMT1, FMT2, GUIDE
(0022)  C-----------------------------------------------------------------
(0023)  C LOCAL DECLARATIONS
(0024)        LOGICAL NOOLD, SAVOLD
(0025)        INTEGER ADBREF(20)
(0026)        INTEGER CODE
(0027)        INTEGER DBNAME(2), RENAME(2)
(0028)        DIMENSION IRSP(1)
(0029)  C-----------------------------------------------------------------
(0030)  C
(0031)  C SET UP HOLLERITH ARRAYS AND CONSTANTS.
(0032)  C
(0033)        RINF= 1.E37
(0034)        CRT= 1
(0035)        OLDDB= 5
(0036)        NEWDB= 6
(0037)        NN= 13
(0038)        NV= 11
(0039)        CALL SETHOL( NAM(1, 1), 52, 52H ENTER MINIMUM CL (WT PCT) ....
(0040)       *.................. )
(0041)        VLU(2, 1)= 0.
(0042)        VLU(3, 1)= 100.
(0043)        CALL SETHOL( NAM(1, 2), 52, 52H ENTER MAXIMUM CL (WT PCT) ....
(0044)       *.................. )
(0045)        VLU(2, 2)= 0.
(0046)        VLU(3, 2)= 100.
(0047)        CALL SETHOL( NAM(1, 3), 52, 52H ENTER TEMPERATURE-COMPOSITION
(0048)       *PE (DEG C/PCT) .. )
(0049)        VLU(2, 3)= -RINF
(0050)        VLU(3, 3)= 0.
(0051)        CALL SETHOL( NAM(1, 4), 52, 52H ENTER EQUILIBRIUM PARTITION RA
(0052)       * .................. )
(0053)        VLU(2, 4)= 0.
(0054)        VLU(3, 4)= 1.
(0055)        CALL SETHOL( NAM(1, 5), 52, 52H ENTER EUTECTIC COMPOSITION (WT
(0056)       *T) ............... )
(0057)        VLU(2, 5)= 0.
(0058)        VLU(3, 5)= 100.
(0059)        CALL SETHOL( NAM(1, 6), 52, 52H ENTER EUTECTIC TEMPERATURE (DE
(0060)       *) ................ )
```

```
C
(0061)          VLU(2, 6)= -RINF
(0062)          VLU(3, 6)=  RINF
(0063)          CALL SETHOL( NAM(1, 7), 52, 52H  ENTER COMPOSITION-DENSITY SLOPE
(0064)         *GM/CM**3)/PCT ... )
(0065)          VLU(2, 7)= -RINF
(0066)          VLU(3, 7)=  RINF
(0067)          CALL SETHOL( NAM(1, 8), 52, 52H  ENTER PRIMARY PHASE SOLID DENSI
(0068)         * (GM/CM**3) ..... )
(0069)          VLU(2, 8)= 0.
(0070)          VLU(3, 8)= RINF
(0071)          CALL SETHOL( NAM(1, 9), 52, 52H  ENTER EUTECTIC LIQUID DENSITY (
(0072)         */CM**3) ......... )
(0073)          VLU(2, 9)= 0.
(0074)          VLU(3, 9)= RINF
(0075)          CALL SETHOL( NAM(1,10), 52, 52H  ENTER EUTECTIC SOLID DENSITY (C
(0076)         *CM**3) ........... )
(0077)          VLU(2,10)= 0.
(0078)          VLU(3,10)= RINF
(0079)          CALL SETHOL( NAM(1,11), 52, 52H  ENTER VISCOSITY (GM/(CM*SEC)) .
(0080)         *................. )
(0081)          VLU(2,11)=0.
(0082)          VLU(3,11)=RINF
(0083)          CALL SETHOL( DBNAME, 8, 8HM1.D.B   )
(0084)          CALL SETHOL( RENAME, 8, 8HM1.TDB   )
(0085)          CALL SETHOL( FERROR, 28, 28H(22H***** INVALID RESPONSE)  )
(0086)          CALL SETHOL( FBOUND, 56, 56H(29H***** VALUE MUST BE IN RANGE,1PE
(0087)         *.3,3H TO,1PE10.3)  )
(0088)          CALL SETHOL( FA4, 4, 4H(A4) )
(0089)          CALL SETHOL( F80CH, 8, 8H(20A4)   )
(0090)          CALL SETHOL( FMT1, 20, 20H(A4,6X,A4,6X,2E10.3) )
(0091)          CALL SETHOL( FMT2, 8, 8H(8E10.3) )
(0092)          CALL SETHOL( GUIDE, 80, 80H.........0.........0.........0.......
(0093)         *0.........0.........0.........0.........0 )
(0094)  C
(0095)  C INITALIZE TEKTRONIX TERMINAL CONTROL SYSTEM.
(0096)  C (USED ONLY FOR NEWPAG.)
(0097)  C
(0098)           CALL INITT(1200)
(0099)           CALL TERM(3,1024)
(0100)           CALL CHRSIZ(4)
(0101)  C
(0102)  C CHECK FOR EXISTANCE OF OLD DATA BASE.
(0103)  C
(0104)          CALL FILES( 1, DBNAME, DUM, DUM, CODE )
(0105)          NOOLD= (CODE.NE.0)
(0106)          IF (NOOLD) CALL MSG( 48, 48HNO OLD DATA BASE. BEGIN BUILDING NE
(0107)         *ATA BASE.  )
(0108)          IF ( NOOLD ) GO TO 310
(0109)  C
(0110)  C CHANGE NAME OF OLD DATABASE.
(0111)  C IF USER WANTS TO SAVE OLD, ASK FOR NAME.
(0112)  C OTHERWISE USE TEMPORARY NAME RENAME FOR OLD DATA BASE, AND DELETE I
(0113)  C AFTER NEW IS WRITTEN.
(0114)  C
(0115)   200   CALL MSG( 60, 60HDO YOU WANT TO SAVE A COPY OF THE OLD DATA BASE
(0116)         * (Y OR N)   )
(0117)          SAVOLD= .FALSE.
(0118)          IRSP(1)= 4H
(0119)          READ (CRT,FA4,ERR=230) IRSP
(0120)          IF ( (IRSP(1).EQ.4HN   ).OR.(IRSP(1).EQ.4HNO  ) ) GO TO 260
```

```
C
      IF ( (IRSP(1).EQ.4HY    ).OR.(IRSP(1).EQ.4HYE   ).OR.
     *     (IRSP(1).EQ.4HYES ) ) GO TO 250
  230 WRITE (CRT,FERROR)
      GO TO 200
C
  250 SAVOLD= .TRUE.
      CALL MSG( 44, 44HENTER COPY FILE NAME (UP TO 6 CHARACTERS)     )
      READ (CRT,F80CH) RENAME
      CALL FILES( 1, RENAME, DUM, DUM, CODE )
      IF (CODE.NE.0) GO TO 270
      CALL MSG( 24, 24HNAME ILLEGAL OR IN USE   )
      GO TO 250
C
  260 CALL FILES( 5, RENAME, DUM, DUM, CODE )
  270 CALL FILES( 6, DBNAME, DUM, RENAME, CODE )
      IF (CODE.NE.0) GO TO 3000
C
C OPEN OLD AND NEW DATA BASES AND GET IDENTIFIER FOR NEW.
C
      CALL FILES( 2, RENAME, OLDDB, DUM, CODE )
      IF ( CODE.NE.0 ) GO TO 3000
      READ (OLDDB,F80CH,END=480,ERR=490) ADBREF
      CALL MSG( 28, 28HOLD DATA BASE IDENTIFIER IS: )
      CALL MSG( 80, ADBREF )
      CALL MSG( 4, 4H     )
  310 CALL FILES( 3, DBNAME, NEWDB, DUM, CODE )
      IF ( CODE.NE.0 ) GO TO 3000
      CALL MSG( 52, 52HENTER NEW DATA BASE IDENTIFIER (UP TO 80 CHARA
     *RS) )
      CALL MSG( 80, GUIDE )
      READ (CRT,F80CH) ADBREF
      WRITE (NEWDB,F80CH) ADBREF
      IF (NOOLD) GO TO 500
C
C READ EACH DATA SET FROM OLD DATA BASE, MODIFY IT, AND WRITE IT
C OUT TO NEW DATA BASE.
C
  400 READ (OLDDB,FMT1,END=500,ERR=490)  SOLVNT, SOLUTE, VLU(1,1),
     *                                   VLU(1,2)
      READ (OLDDB,F80CH,END=480,ERR=490) ALREF
      READ (OLDDB,FMT2,END=480,ERR=490)  (VLU(1,IV),IV=3,6)
      READ (OLDDB,FMT2,END=480,ERR=490)  (VLU(1,IV),IV=7,10)
      READ (OLDDB,FMT2,END=480,ERR=490)  VLU(1,11)
C
      CALL MODWRI
      GO TO 400
C
  480 CALL MSG( 44, 44H*****UNEXPECTED END-OF-FILE ON OLD DATA BASE )
      CALL MSG( 32, 32H*****SKIPPING TO DATA SET INPUT   )
      GO TO 500
  490 CALL MSG( 36, 36H*****FORMAT ERROR IN OLD DATABASE   )
      CALL MSG( 32, 32H*****SKIPPING TO DATA SET INPUT   )
C
C INPUT NEW DATA SET, MODIFY IT IF NECESSARY, AND WRITE IT
C OUT TO NEW DATA BASE.
C
  500 CALL NEWPAG
      CALL MSG(48, 48HENTER NAME OF SOLVENT (UP TO 4 CHARACTERS), OR
      CALL MSG(48, 48H      Q     TO TERMINATE DATA SET SPECIFICATION
      READ (CRT,FA4) SOLVNT
```

```
C
(0181)        IF (SOLVNT.EQ.4HQ   ) GO TO 600
(0182)  C
(0183)        CALL MSG( 44, 44HENTER NAME OF SOLUTE (UP TO 4 CHARACTERS)    )
(0184)        READ (CRT,FA4) SOLUTE
(0185)  C
(0186)        CALL MSG( 56, 56HENTER ALLOY REFERENCE (5 LINES OF UP TO 80 CHA
(0187)       *TERS)    )
(0188)        CALL MSG( 80, GUIDE ).
(0189)        READ (CRT,F80CH) ALREF
(0190)  C
(0191)        DO 580 IV=1,NV
(0192)   520  CALL MSG( 4*NN, NAM(1,IV) )
(0193)        READ (CRT,*,ERR=560) VLU(1,IV)
(0194)        IF ( (VLU(1,IV).GE.VLU(2,IV)) .AND.
(0195)       *     (VLU(1,IV).LE.VLU(3,IV)) ) GO TO 580
(0196)        WRITE (CRT,FBOUND) VLU(2,IV), VLU(3,IV)
(0197)        GO TO 520
(0198)   560  WRITE (CRT,FERROR)
(0199)        GO TO 520
(0200)   580  CONTINUE
(0201)  C
(0202)        CALL MODWRI
(0203)        GO TO 500
(0204)  C
(0205)  C CLOSE NEW DATA BASE, DELETE OLD IF IT WAS NOT TO BE SAVED.
(0206)  C
(0207)   600  CALL FILES( 4, DBNAME, NEWDB, DUM, CODE )
(0208)        IF (NOOLD) CALL EXIT
(0209)        CALL FILES( 4, RENAME, OLDDB, DUM, CODE )
(0210)        IF (.NOT.SAVOLD) CALL FILES( 5, RENAME, DUM, DUM, CODE )
(0211)        CALL EXIT
(0212)  C
(0213)   3000 CALL MSG( 20, 20H****FILE USAGE ERROR )
(0214)        CALL EXIT
(0215)  C
(0216)        END
PROGRAM SIZE:   PROCEDURE - 003432    LINKAGE - 000341    STACK - 000022
0000 ERRORS [<.MAIN.>FTN-REV15.3]
```

SUBROUTINE MODWRI

```
(0217)          SUBROUTINE MODWRI
(0218)    C
(0219)    C MODIFY INDIVIDUAL ELEMENTS OF DATA SET, AND
(0220)    C WRITE DATA SET OUT TO NEW DATA BASE.
(0221)    C
(0222)    C-----------------------------------------------------------------------
(0223)    C COMMON BLOCKS
(0224)          COMMON       CRT, OLDDB, NEWDB,
(0225)         *             NAM(13,15), NN, VLU(3,15), NV,
(0226)         *             SOLVNT, SOLUTE, ALREF(20,5),
(0227)         *             FERROR(7), FBOUND(14), FA4(1), FBDCH(2),
(0228)         *             FMT1(5), FMT2(2), GUIDE(20)
(0229)          INTEGER      CRT, OLDDB, NEWDB,
(0230)         *             SOLVNT, SOLUTE, ALREF,
(0231)         *             FERROR, FBOUND, FA4, FBDCH,
(0232)         *             FMT1, FMT2, GUIDE
(0233)    C-----------------------------------------------------------------------
(0234)    C LOCAL DECLARATIONS
(0235)          DIMENSION IRSP(1)
(0236)    C-----------------------------------------------------------------------
(0237)    C
(0238)    C CLEAR SCREEN AND WRITE CURRENT VALUES.
(0239)    C
(0240)     200  CALL NEWPAG
(0241)    C
(0242)          WRITE (CRT,220) SOLVNT, SOLUTE
(0243)     220  FORMAT( 15H   1   SOLVENT:, 5X, A4,
(0244)         *       /15H   2   SOLUTE: , 5X, A4 )
(0245)    C
(0246)          WRITE (CRT,230) ALREF
(0247)     230  FORMAT( 24H   3   ALLOY REFERENCE: , 20A4,
(0248)         *       / (24X, 20A4) )
(0249)    C
(0250)          NV3= NV+3
(0251)          WRITE (CRT,240) ( IV, (NAM(IN,IV-3),IN=3,NN), VLU(1,IV-3),
(0252)         *                  IV=4,NV3 )
(0253)     240  FORMAT( I4, 3X, 11A4, 5H.... , 1PE12.5 )
(0254)    C
(0255)    C ALLOW USER TO MAKE CHANGES.
(0256)    C
(0257)     300  CALL MSG( 4, 4H      )
(0258)          CALL MSG( 4, 4H      )
(0259)          CALL MSG( 4, 4H      )
(0260)          CALL MSG( 68,68HENTER C TO COPY THIS DATA SET AS IT STANDS INTO
(0261)         *E NEW DATA BASE,    )
(0262)          CALL MSG( 56,56H      O TO OMIT THIS DATA SET FROM THE NEW DATA
(0263)         *SE, OR )
(0264)          CALL MSG( 28,28H      ITEM NUMBER TO CHANGE. )
(0265)          IRSP(1)= 4H
(0266)          READ (CRT,FA4,ERR=340) IRSP
(0267)          IF (IRSP(1).EQ.4HC    ) GO TO 500
(0268)          IF (IRSP(1).EQ.4HO    ) RETURN
(0269)          IF (IRSP(1).EQ.4H     ) GO TO 340
(0270)          DECODE( 4, *, IRSP, ERR=340 ) ITEM
(0271)          IF ( (ITEM.LT.NV+4) .AND. (ITEM.GT.0) ) GO TO 350
(0272)     340  WRITE (CPT,FERROR)
(0273)          GO TO 300
(0274)    C
(0275)     350  IF ( ITEM.GT.3 ) GO TO 410
(0276)          GO TO (360,370,380), ITEM
```

SUBROUTINE MODWRI

```
(0277)    360  CALL MSG( 36, 36HENTER SOLVENT (UP TO 4 CHARACTERS)    )
(0278)         READ (CRT,FA4) SOLVNT
(0279)         GO TO 200
(0280)    370  CALL MSG( 36, 36HENTER SOLUTE  (UP TO 4 CHARACTERS)    )
(0281)         READ (CRT,FA4) SOLUTE
(0282)         GO TO 200
(0283)    380  CALL MSG( 56, 56HENTER ALLOY REFERENCE (5 LINES OF UP TO 80 CHA
(0284)        *TERS)    )
(0285)         CALL MSG( 80, GUIDE )
(0286)         READ (CRT,F80CH) ALREF
(0287)         GO TO 200
(0288)    410  IV=ITEM-3
(0289)         CALL MSG( 4*NN, NAM(1,IV) )
(0290)         READ (CRT,*,ERR=430) VLU(1,IV)
(0291)         IF ( (VLU(1,IV).GE.VLU(2,IV)) .AND.
(0292)        *     (VLU(1,IV).LE.VLU(3,IV)) ) GO TO 200
(0293)         WRITE (CRT,FBOUND) VLU(2,IV), VLU(3,IV)
(0294)         GO TO 410
(0295)    430  WRITE (CRT,FERROR)
(0296)         GO TO 410
(0297)    C
(0298)    C WRITE CURRENT VALUES OUT TO NEW DATA BASE.
(0299)    C
(0300)    500  WRITE (NEWDB,FMT1)  SOLVNT, SOLUTE, VLU(1,1), VLU(1,2)
(0301)         WRITE (NEWDB,F80CH) ALREF
(0302)         WRITE (NEWDB,FMT2)  (VLU(1,IV),IV=3,6)
(0303)         WRITE (NEWDB,FMT2)  (VLU(1,IV),IV=7,10)
(0304)         WRITE (NEWDB,FMT2)  VLU(1,11)
(0305)         RETURN
(0306)         END
PROGRAM SIZE:   PROCEDURE - 001460    LINKAGE - 000134    STACK - 000020
0000 ERRORS [<MODWRI>FTN-REV15.3]
```

```
      SUBROUTINE FILES( NTRY, FNAME, FUNIT, CNAME, CODE )

(0307)        SUBROUTINE FILES( NTRY, FNAME, FUNIT, CNAME, CODE )
(0308)  C
(0309)  C PERFORMS FILE OPERATIONS USING PRIME SYSTEM ROUTINES.
(0310)  C
(0311)  C-----------------------------------------------------------------
(0312)  C     NTRY     (INPUT)     1  CHECK EXISTANCE OF FILE FNAME IN CURRENT UF
(0313)  C                             RETURNS CODE= 0 IF IT EXISTS,
(0314)  C                                           NONZERO OTHERWISE.
(0315)  C                          2  OPEN FILE FNAME ON FORTRAN LOGICAL UNIT
(0316)  C                             FUNIT TO READ.
(0317)  C                          3  OPEN FILE FNAME ON FORTRAN LOGICAL UNIT
(0318)  C                             FUNIT TO WRITE.
(0319)  C                          4  CLOSE FILE FNAME.
(0320)  C                          5  DELETE FILE FNAME FROM CURRENT UFD.
(0321)  C                          6  CHANGE NAME OF FILE FNAME TO CNAME.
(0322)  C     FNAME    (INPUT)     HOLLERITH FILE NAME. UP TO 6 CHARACTERS.
(0323)                             INTEGER FNAME(2)
(0324)  C     FUNIT    (INPUT)     FORTRAN LOGICAL UNIT.
(0325)                             INTEGER FUNIT
(0326)  C     CNAME    (INPUT)     NEW FILE NAME FOR ENTRY 6.
(0327)                             INTEGER CNAME(2)
(0328)  C     CODE     (OUTPUT)    ERROR CODE.
(0329)  C                          CODE= 0 IF OPERATION SUCCESSFUL.
(0330)  C                              NONZERO OTHERWISE.
(0331)                             INTEGER CODE
(0332)  C-----------------------------------------------------------------
(0333)  C SYSTEM SUPPLIED PARAMETERS
(0334)  C SYSCOM>KEYS.F       MNEMONIC KEYS FOR FILE SYSTEM (FTN)     31 MAY, 1
(0334)        NOLIST
(0335)  C-----------------------------------------------------------------
(0336)  C LOCAL DECLARATIONS
(0337)        INTEGER*2 INAME, KODE, PUNIT, SPAR,
(0338)       *          NTEXT
(0339)        INTEGER TEXT(2,6)
(0340)        DATA TEXT  / 4HFIND,4H    , 4HOPEN,4H    ,
(0341)       *             4HOPEN,4H    , 4HCLOS,4HE   ,
(0342)       *             4HDELE,4HTE  , 4HRENA,4HME  /,
(0343)       *     NTEXT / 8 /
(0344)        DATA INAME / 6 /
(0345)  C-----------------------------------------------------------------
(0346)  C
(0347)        CODE= 0
(0348)        IF (NTRY.EQ.6) GO TO 500
(0349)        SPAR= K$EXST
(0350)        IF (NTRY.EQ.2) SPAR= K$READ
(0351)        IF (NTRY.EQ.3) SPAR= K$WRIT
(0352)        IF (NTRY.EQ.4) SPAR= K$CLOS
(0353)        IF (NTRY.EQ.5) SPAR= K$DELE
(0354)  C
(0355)        PUNIT= FUNIT- 4
(0356)        IF ( (NTRY.EQ.1).OR.(NTRY.EQ.5) ) PUNIT= 0
(0357)        ITYPE= 0
(0358)  C
(0359)        CALL SRCH$$( SPAR, FNAME, INAME, PUNIT, ITYPE, KODE )
(0360)        CODE= KODE
(0361)  C
(0362)        IF (NTRY.EQ.1) RETURN
(0363)        IF ( (NTRY.EQ.5) .AND. (CODE.EQ.15) ) RETURN
(0364)        CALL ERRPR$( K$IRTN, KODE, TEXT(1,NTRY), NTEXT, FNAME, INAME)
(0365)  C
```

```
      SUBROUTINE FILES( NTRY, FNAME, FUNIT, CNAME, CODE )

(0366)            RETURN
(0367)  C
(0368)     500    CALL CNAMSS( FNAME, INAME, CNAME, INAME, KODE )
(0369)            CODE= KODE
(0370)            CALL ERPPRS( KSIRTN, KODE, TEXT(1,NTRY), NTEXT, FNAME, INAME)
(0371)            RETURN
(0372)  C
(0373)            END
PROGRAM SIZE:   PROCEDURE - 000266    LINKAGE - 000071    STACK - 000036
0000 ERRORS [<FILES >FTN-REV19.3]
```

```
SUBROUTINE MSG ( NCH, MESSGE )


(0374)        SUBROUTINE MSG ( NCH, MESSGE )
(0375)  C
(0376)  C DISPLAYS MESSAGE ON CRT.
(0377)  C
(0378)  C-------------------------------------------------------------------
(0379)  C     NCH      (INPUT)    NUMBER OF CHARACTERS IN MESSAGE.
(0380)  C     MESSGE   (INPUT)    MESSAGE TO DISPLAY.
(0381)                            DIMENSION MESSGE(NCH)
(0382)  C-------------------------------------------------------------------
(0383)  C LOCAL DECLARATIONS
(0384)        INTEGER CRT
(0385)        DATA CRT/1/, NCW/4/
(0386)  C-------------------------------------------------------------------
(0387)  C
(0388)        NN= NCH/NCW
(0389)        IF (NN*NCW .LT. NCH) NN=NN+1
(0390)        WRITE (CRT,100) (MESSGE(IN),IN=1,NN)
(0391)   100  FORMAT ( 20A4 )
(0392)  C
(0393)        RETURN
(0394)        END
PROGRAM SIZE:   PROCEDURE - 000101    LINKAGE - 000037    STACK - 000022
0000 ERRORS [<MSG   >FTN-REV15.3]
```

SUBROUTINE SETHOL (ARRAY, NCH, HOLCON)

```
(0395)        SUBROUTINE SETHOL ( ARRAY, NCH, HOLCON )
(0396)  C
(0397)  C PUTS HOLLERITH CONSTANT INTO ARRAY.
(0398)  C USED TO AVOID THE CRUDE RESTRICTIONS OF THE ANSI
(0399)  C DATA STATEMENT.
(0400)  C
(0401)  C-----------------------------------------------------------------------
(0402)  C ARGUMENT LIST
(0403)  C
(0404)  C     ARRAY     (INPUT)   HOLLERITH STRING WILL BE STORED IN ARRAY.
(0405)                            INTEGER ARRAY (NCH)
(0406)  C     NCH       (INPUT)   NUMBER OF CHARACTERS IN HOLLERITH STRING.
(0407)  C     HOLCON    (INPUT)   HOLLERITH STRING TO STORE.
(0408)                            INTEGER HOLCON (NCH)
(0409)  C-----------------------------------------------------------------------
(0410)  C LOCAL DECLARATIONS
(0411)        DATA NCW/4/
(0412)  C-----------------------------------------------------------------------
(0413)  C
(0414)        NN= NCH/NCW
(0415)        IF ( NN*NCW .LT. NCH ) NN= NN+1
(0416)        DO 100 IN=1,NN
(0417)        ARRAY(IN)= HOLCON(IN)
(0418)   100  CONTINUE
(0419)  C
(0420)        RETURN
(0421)        END
PROGRAM SIZE:   PROCEDURE - 000057    LINKAGE - 000026    STACK - 000024
0000 ERRORS [<SETHOL>FTN-REV15.3]
```